Sensor Based Motion Planning:
The Hierarchical Generalized Voronoi Graph

Thesis by
Howie Choset

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

1996

(Defended March 5, 1996)

Abstract

Sensor based motion planning incorporates sensor information reflecting the state of a robot's environment into its planning process, whereas traditional approaches assume complete prior knowledge of the robot's environment. Recent research has focused on the development and incremental construction of the hierarchical generalized Voronoi graph (HGVG), which is a concise representation of a robot's environment. The HGVG is advantageous in that it lends itself to sensor based construction in a rigorous and provably correct manner. With this approach, a robot can enter an unknown environment, incrementally construct the HGVG, and then use the HGVG for future excursions in the environment. Simulations and experiments validate this approach.

Acknowledgments

Many people have touched my life and influenced my work at Caltech and to each of them, I owe a great debt. My long list of thank you's starts with my advisor, Dr. Joel Burdick, for without whom, my wonderful experiences at Caltech never would have occurred. Although I still do not understand why there is no machine named after Wanky, I am thankful for his guidance (in most things!!), his willingness to talk to me about anything, and his wisdom (see my web page). I wish the best for Joel and his family.

Next, I would like to thank the members of my PhD committee, starting with Dr. Andrew Lewis. He perhaps has had the greatest positive affect on my work and me throughout my years at Caltech; he has been with me for so many things, some of which includes helping me prepare for my candidacy exam, getting me started with nonsmooth analysis, constantly reviewing my work, sharing a day with me in the hospital when I had salmonella, going to Penny's and just being an all around good friend to me.

Some of my fondest memories of Caltech include talking about my research with Dr. Richard Murray in his office. Many times he made time for me and always spoke to me as a colleague. As far as I am concerned, he is my back-up advisor. I would like to thank Dr. Erik Antonsson for being a last minute member of my committee and his kind words towards my work. Also, I would like to thank him for making me so popular with the ladies. Finally, I would like to thank Dr. Perona for showing me during my thesis defense that I have built up a type of intuition that is well suited to the problems I want to solve.

Many other friends at Caltech have influenced my work and me. First, I would like to thank my dear friend Dr. Jim Ostrowski for all the help and birthday surprises he has provided for me over the years; he is a true buddy who always has been gentle and kind. Next is Luis Goncalves for his research discussions, especially the infamous one at Malibu. Also, a thank you should be sent to Dr. Dave Kriegman who ruined by research three years ago (I know where you live, Dave). I am also grateful to Dr. Greg Chirikjian who (1) started snake research, (2) helped me get through my turbulent first year of Caltech, and (3) took me to B.E. so many times. I would also like to thank Will Law for helping me write better and talking science; as long as women walk the face of this Earth, we will have something to talk about. I have also enjoyed my research discussions with Captain Richard Mason. In addition to research help, I would like to thank my fellow New Yorker, Scott Kelley, for exposing me to a culture of many things that are all round. Thank you to Jim Radford and Jonah Harley for their help with the experiments. And last, but not smallest, I would like to thank Bill Goodwine (hereinafter "Franz") for his friendship, his partnership in the weight room, and the many discussions on everything at work (in the weight room, Franz and I just lifted weights).

There have been many people outside of work whose valued friendships kept me going through the trials and tribulations of Caltech. My first new California friend is Carlos Jorquera, who from the very first moment I met him, I knew would be my life-long friend. And there is my Jewish friend Richard Tsuyuki who convinced me I was a good guy; if I had been woman, he would have been mine. Richard is just about the only one I would call Dick and mean it. I would like to thank Fred Farina for forcing me to learn French curses; I am sure it will be useful one day. Finally, I would like to thank Gilead Grossman for being my loyal servant at Venice beach.

I appreciate members of my Tae Kwon Do club for the frequent workouts and keeping me honest. In particular, I am thankful to Dr. Eric Paljug, Bill Williamson, Bobby Johnson, Tim Doyle, Dr. Anthony D'Amico, Jodi Forlizzi and especially Susan Downey. I would also like to thank Eric for following me from Penn, to Caltech and then to Pittsburgh. Also, I am thankful to the boss and fellow New Yorker, Charmaign Boyd, for keeping things smooth; if I ever turn to a life of crime, I want her on my side. Then there is the best officemate I ever had, Brett Slatkin; he gives the best birthday presents I ever received. Also, there is my dear friend Steller Kaoru Oota whom I met just recently, but seem to think is pretty neat. Finally, I would like to thank the constant support I received from my Penn roommates Ron "Wanky" Levinson, Barry "Shlanky" Friedson, Mark "Snooter" Monack, and Jodie "Monack" Lane; I am really sorry that wanky.caltech.edu never came alive (blame

Joel).

I would like to thank the Office of Naval Research for sponsoring this research. Finally, I would like to thank my parents, Malcolm and Tobe, for endowing me with a strong work ethic, without which, I never would have accomplished anything. I would also like to thank my mother for getting stains of all kinds out of my clothing. To my parents, this thesis is dedicated.

Contents

List of Symbols

The symbols are grouped into five categories: Voronoi structures, types of spaces, mathematical notation, distance functions, and miscellaneous structures.

Symbol	Meaning	Definition
VD	Voronoi diagram	Set of points in the plane equidistant to two or more points called site points.
GVD	generalized Voronoi diagram	Set of points in the plane equidistant to two or more convex sets.
VG (RVG)	Voronoi graph (regular Voronoi graph)	Set of points in an m-dimensional Euclidean space equidistant to m or more point sites.
GVG	generalized Voronoi graph	Set of points in an m-dimensional Euclidean space equidistant to m or more convex sets.
GVG^2	second order generalized Voronoi graph	Set of points equidistant to a pair of closest obstacles and $m-1$ second closest obstacles.
HGVG	hierarchical generalized Voronoi graph	The GVG with additional structures termed higher order GVG's.
L^{3}HGVG	linked hierarchical generalized Voronoi graph	The HGVG with additional links.

Symbol	Meaning
$\mathbb{R}^m$	m-dimensional Euclidean Space
$\mathcal{W}$	Work space
C_i	Obstacle i (assumed to be convex)
$\mathcal{FS}$	Free space

Symbol	Meaning
dim	Dimension
codim	Co-dimension
nbhd	Neighborhood (of a point)
int	Interior (of a set)
cl	Closure (of a set)
∂	Boundary (of a set) *or* generalized gradient (of a function)
$\bigcap$	Intersection
$\bigcup$	Union
$\setminus$	Subtraction
Co	Convex hull

Sym.	Meaning	Definition
d_i	single object distance function	Distance between a point and a convex obstacle (which may *or* may not be occluded).
D	multi-object distance function	Distance between a point and the nearest obstacle.
∇d_i	gradient of single object distance function	A unit vector pointing away from the nearest point on the obstacle.
∂D	generalized gradient of the multi-object distance function	The convex hull of the single object gradient vectors whose distance is the same as the multi-object distance function.
d_i^X	X-distance function *or* X-ray distance function	Distance between a point and a convex obstacle (which may be occluded from the point).
d_i^S	S-distance function *or* slanted distance function	Distance between a point and the nearest line of sight point in the obstacle.
d_i^V	V-distance function *or* visible distance function	Distance between a point and the nearest line of sight point in the interior of the obstacle line of sight points.

Symbol	Meaning	Distance relationships
$\mathcal{SS}_{ij}$	two-equidistant surjective surface	$d_i(x) = d_j(x)$
$\mathcal{SS}_{i_1 \ldots i_k}$	k-equidistant surjective surface	$d_{i_1}(x) = \cdots = d_{i_k}(x)$
$\mathcal{F}_i$	generalized Voronoi region	$d_i(x) \leq d_h(x) \, \forall h$
$\mathcal{F}_{ij}$	two-equidistant face *or* generalized Voronoi face	$d_i(x) = d_j(x) \leq d_h(x) \; \forall h$
$\boldsymbol{\mathcal{F}}^2$	two-Voronoi set *or* GVD	
$\mathcal{F}_{i_1 \ldots i_k}$	k-equidistant face	$d_{i_1}(x) = \cdots = d_{i_k}(x) \leq d_h(x) \; \forall h$
$\boldsymbol{\mathcal{F}}^k$	k-Voronoi set	
$\mathcal{F}_{i_1 \ldots i_m}$	m-equidistant face *or* GVG edge	$d_{i_1}(x) = \cdots = d_{i_m}(x)$
$\mathcal{F}_{i_1 \ldots i_{m+1}}$	$m+1$-equidistant face *or* GVG vertex	$d_{i_1}(x) = \cdots = d_{i_{m+1}}(x)$
$\mathcal{F}_k\big\vert_{\mathcal{F}_{ij}}$	second order generalized Voronoi region	$d_i(x) = d_j(x) \leq d_k(x) \leq d_h(x) \; \forall h$
$\mathcal{F}_{i_1 \ldots i_{m-1}}\big\vert_{\mathcal{F}_{ij}}$	GVG^2 equidistant edge	$d_i(x) = d_j(x) \leq d_{i_1}(x) = \cdots = d_{i_{m-1}}(x) \leq d_h(x) \; \forall h$
$C_{i_1 \ldots i_k}$	k-boundary face	$d_{i_1}(x) = \cdots = d_{i_k}(x) = 0$
$C_{i_1 \ldots i_m}$	boundary edge *or* m-boundary face	$d_{i_1}(x) = \cdots = d_{i_m}(x) = 0$
$FC_{i_1 \ldots i_k}$	floating k-boundary face	$d_{i_1}(x) = \cdots = d_{i_k}(x) \leq d_h(x) \; \forall h$
$V_{kl}\big\vert_{\mathcal{F}_{ij}}$	occluding two-face	$d_i(x) = d_j(x) < d_k(x) < d_l(x) < d_h(x) \; \forall h$ *or* $d_i(x) = d_j(x) < d_l(x) < d_k(x) < d_h(x) \; \forall h$

Chapter 1

Introduction

The sensor based motion planning problem is to find a collision-free path for a robot only using environmental information provided by sensors. This problem is an extension to *classical motion planning*, where environmental information is already available to the robot prior to the planning event. There already exists many provably correct algorithms for classical motion planning, but although many algorithms exist for sensor based motion planning, very few of them are provably correct. Existing provably correct algorithms for sensor based motion planning are limited to planar problems. This thesis presents a provably correct sensor based motion planning algorithm that is applicable to dimensions greater than two.

1.1 Motivation and Problem Statement

Sensor based motion planning is important for realistic deployment of robots because: (1) the robot may have no a priori knowledge of the world; (2) the robot may have only a coarse knowledge of the world; (3) the robot's knowledge of its environment may be inaccurate; and (4) the robot's world may be subject to unexpected occurrences or rapidly changing situations. The robot must rely on its sensors in order to effectively function in these situations.

Two canonical problems in sensor based motion planning are addressed in this thesis: finding a collision-free path to a goal location and mapping a bounded environment with a systematic exploration procedure. In both problems, the robot

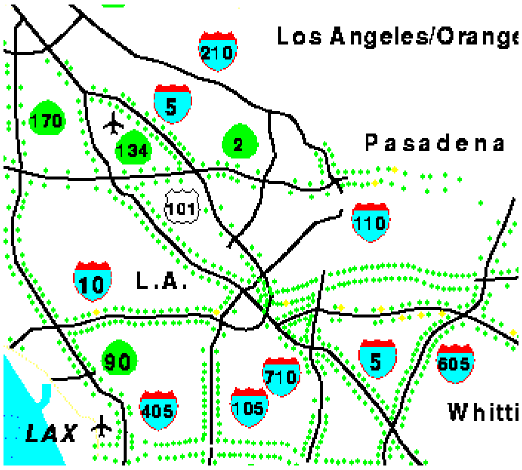

Figure 1.1 Traffic Map of Los Angeles on October 5, 1995.

has no a priori knowledge of the environment, but always knows its current location (e.g., it may have an on-board dead reckoning system) and its sensor range is sufficient.

The solution to the mapping problem automatically supplies a solution to the collision-free path planning problem in a bounded environment, and so we will focus our attention on the mapping problem. To this end, we construct a *roadmap*: a network of one-dimensional curves that concisely represents the geometry of a robot's environment. A roadmap is analogous to a system of freeways (See Figure 1.1). Using a freeway system, travel between two points is accomplished by driving directly to the freeway, traversing a sequence of freeways to an exit near the destination, and then proceeding directly to the destination. The *hierarchical generalized Voronoi graph* (HGVG), defined in this thesis, is a roadmap that can be incrementally constructed using line of sight sensor data.

A sensor based planner should have three important qualities: (1) it can create a concise representation of the environment, (2) it constructs a map or motion plan in an incremental manner, and (3) it is amenable to realistic sensors. Since the HGVG is a roadmap, it already is a concise representation of an environment. The HGVG is defined in terms of line of sight distance information which gives rise to an incremental construction procedure that is described in this work. The incremental construction procedure relies on line of sight range information which can be readily obtained from realistic sensors.

1.2 Relation to Previous Work

Any practical robot motion planning algorithm must be *complete*, i.e., find the goal if it is reachable or map the entire environment in finite time. More sophisticated algorithms allow for moving obstacles, and various “soft” optimization criteria, such as maximizing the clearance between the robot and obstacles, and minimizing the distance traveled by the robot during the quest for the goal.

1.2.1 Classical Motion Planning

In the classical robot motion planning problem, the robot has complete knowledge of the world's geometry is available prior to the start of the planning event. Throughout this work, we assume the robot is modeled as a point operating in a subset $\mathcal{W}$ of an m-dimensional Euclidean space. When m is two or three, $\mathcal{W}$ could be either a configuration space or a workspace, but when $m > 3$, $\mathcal{W}$ is a configuration space which is diffeomorphic to $\mathbb{R}^m$. The work space $\mathcal{W}$ is populated by obstacles $C_1, \ldots, C_n$. In later sections, we assume that obstacles are closed convex sets and that non-convex obstacles are modeled as the union of the convex sets. The $\mathcal{W}$ is populated with a set of convex obstacle, $\{C_1, \ldots, C_n\}$. The set of points where the robot is free to move is called the *free space*, and is defined as $\mathcal{FS} = \mathcal{W} \setminus \bigcup_{i=1}^{i=n} C_i$. See Figure 1.2.

Assumption 1.1 (Boundedness Assumption) The robot operates in a bounded connected subset of the free space $\mathcal{FS}$. This subset is bounded by obstacles.

When Assumption 1.1 is upheld, $n \geq m + 1$. For example, when $m = 3$, the minimum number of convex obstacles to bound a subset of the $\mathcal{FS}$ is four. Also note that when Assumption 1.1 is upheld, although the robot is operating in a bounded connected subset of $\mathcal{FS}$, the freespace $\mathcal{FS}$ may be unbounded.

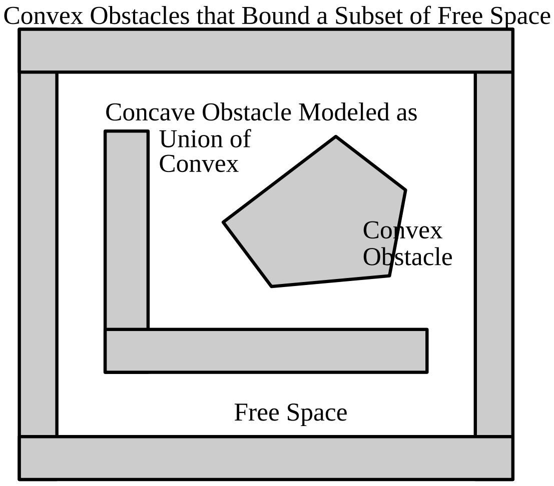

Figure 1.2 The robot operates in a bounded subset of the free space. Concave obstacles are modeled as the union of convex obstacles.

Three classes of complete classical motion planning schemes have been developed: potential field approaches, roadmap methods, and cellular decomposition (Latombe, [28]).

Potential Fields

In the potential field approach (Khatib, [25]; Latombe, [28]), the robot is modeled as a particle acting under the influence of a potential function $U\colon \mathbb{R}^m \to \mathbb{R}$ that encodes information about the environment such as the goal and obstacle location. Typically, these potential functions have two parts: an attractive potential, $U_{att}\colon \mathbb{R}^m \to \mathbb{R}$, and a repulsive potential, $U_{rep}\colon \mathbb{R}^m \to \mathbb{R}$. The attractive potential is normally a quadratic function with its global minima (only local minima) at the goal, thereby guiding the robot towards the goal. The repulsive potential is inversely proportional to distance to obstacles, thereby steering robots away from obstacles. The potential function is the sum of the attractive and repulsive potential functions, i.e., $U = U_{att} + U_{rep}$, and motion planning is effected by gradient descent of U.

The advantage that this approach has over others is that it extends nicely into higher dimensions, and can produce some very efficient solutions. Another advantage of the potential field approach is that a robot can compute simple potential functions using local information; it is this feature which makes potential fields amenable to sensor based planning. Nevertheless, a major disadvantage of this approach is that the robot may get trapped in a local minima different from the local minima associated with the goal. There are methods for dealing with this problem, but either they are restrictive or require too much information about the environment (Rimon and Koditschek, [39]).

Roadmap Methods

Definition 1.2 (Roadmap) A *roadmap* R is the union one-dimensional curves such that for all q_{start} and q_{goal} in $\mathcal{FS}$, there exists a path between q_{start} and q_{goal} if and only if

1. there exists a path from $q_{start} \in \mathcal{FS}$ to some $q'_{start} \in R$ (*accessibility*),
2. there exists a path from $q_{goal} \in \mathcal{FS}$ to some $q'_{goal} \in R$ (*departibility*), and
3. there exists a path in R between q'_{start} and q'_{goal} (*connectivity*).

Using a roadmap (Canny, [6]; Latombe, [28]), the planner can construct a path between any two points in a connected component of the robot's free space by first finding a collision free path onto the roadmap (accessibility), traversing the roadmap to the vicinity of the goal (connectivity), and then constructing a collision free path from a point on the roadmap to the goal (departability). Three primary types of roadmaps have been developed: *visibility graph*, *retraction*, and *silhouette*.

Visibility Graph. The nodes of the *visibility graph* (Latombe, [28]; Lozano-Perez and Wesley, [30]) include the start location, the goal location, and all the vertices of the obstacles (or configuration space obstacles). The edges of the visibility graph are straight line segments that connect two nodes without penetrating the *interior* of the obstacles (or configuration space obstacles). An edge of a polygon obstacle also serves as an edge on the visibility graph because it does not intersect the interior of the obstacle. Using the standard two-norm, the visibility graph can be searched for the shortest path in $\mathbb{R}^2$ (Figure 1.3), but is not guaranteed to find the shortest path in dimensions greater than two.

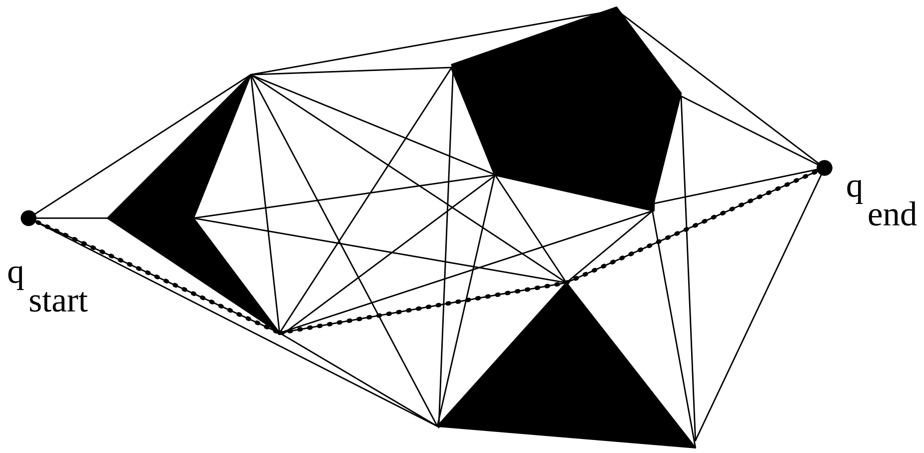

Figure 1.3 Thin solid lines delineate the edges of the visibility graph for the three obstacles represented as filled in polygons. The thick dotted line represents the shortest path between the start and goal.

Retraction. A *retraction* (Ó'Dúnlaing and Yap, [34]) is a continuous mapping of the free space $\mathcal{FS}$ onto a one-dimensional subset of $\mathcal{FS}$. The image of the retraction is the roadmap. A popular retraction in motion planning is termed the *Voronoi diagram.* This structure is a graph whose edges are the set of points whose minimal distance to the boundary of $\mathcal{FS}$ occurs at two distinct points, and whose nodes are the points where edges meet (and terminate). Typically, the nodes are the set of points whose minimal distance to the boundary of $\mathcal{FS}$ set occurs at least three distinct points. Traditionally, this structure was developed for planar motion planning.

Accessibility (resp. departability) is achieved by drawing through the start (resp. goal) a line that is perpendicular to the closest obstacle to the start (resp. goal). This line is guaranteed to intersect the diagram. Let this be point be labeled q'_{start} (resp. q'_{goal}). If a path exists between the start and goal locations, then the connectivity property guarantees a path exists in the Voronoi diagram between q'_{start} and q'_{goal}. Hence, remainder of the motion planning process is reduced to a graph search which connects q'_{start} and q'_{goal} in the graph. See Figure 1.4. This algorithm only works in the planar case.

Originally, the Voronoi diagram applied to various computational geometry

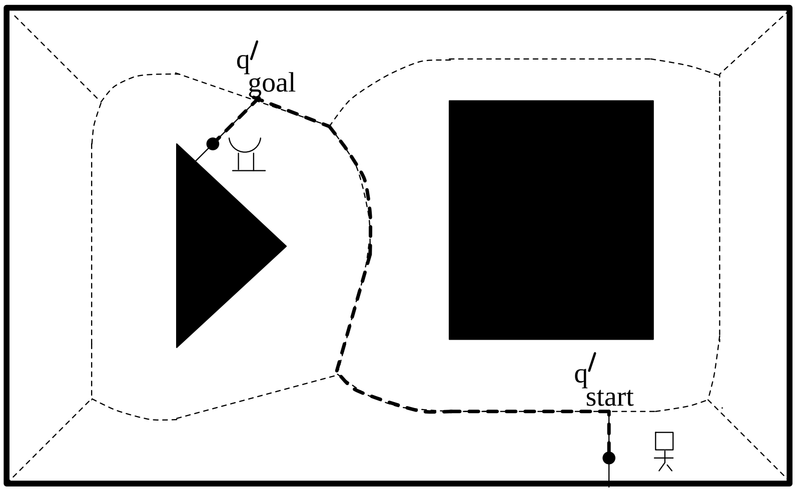

Figure 1.4 The dotted line delineates the generalized Voronoi diagram for a planar environment surrounded by a rectangular enclosure and a triangle and square in its interior. The start (little man) and goal (cup of water) are connected by the thick dotted line.

problems limited to a planar environment populated with a set of isolated points, termed sites. The original Voronoi diagram is the set of points with minimal distance to two or more sites. Although named for the Russian mathematician, Georgii Feodosevich Voronoi who lived from 1868 until 1908, the Voronoi diagram was first studied by DesCartes in 1644. The Voronoi diagram was first applied to robotics in Rowat's work on computer vision [40]. In this and other works, the sites were sets, instead of points. In such cases, in keeping with the Voronoi diagram literature, we term this structure the *generalized Voronoi diagram* in order to distinguish it from the original definition of the Voronoi diagram.

The advantage of the generalized Voronoi diagram is that it prescribes a path of maximal clearance from the obstacles, thereby allowing for safer pathways. The disadvantage of this approach is that it does not directly extend to work spaces whose dimension is greater than two. A major contribution of this work is the development of analogous methods that are applicable to these higher dimensional work spaces.

Silhouette Method. A drawback of the generalized Voronoi diagram is that it is limited to planar motion planning problems. Canny [6] introduced a roadmap approach that is applicable to work space and configuration spaces whose dimension is greater than two. A hyperplane, termed a *slice*, is swept through the work space or a parameterization of configuration space, populated with obstacles. Typically, the slice is perpendicular to a coordinate axis. On each slice, extremal points of some function (e.g., the distance to the closest obstacle) are determined. As the slice is swept, these extremal points trace out curves termed *silhouette curves.* In general, these curves are not guaranteed to be connected; in fact, at points, termed *critical points*, new silhouette curves appear or disappear. At these points, the algorithm is recursively invoked on the slice that passes through the critical points. The recursion terminates when the slice is one-dimensional. The union of the silhouette curves is a connected roadmap.

The *Opportunistic Path Planner* (OPP) of Canny and Lin [8], [9] is an example of a silhouette method in which local maxima of a continuous and differentiable potential function, restricted to a slice, are traced out as a slice sweeps through the work space or a parameterization of configuration space. Canny and Lin suggest that the Euclidean distance function between a given point (which represents the robot's configuration) on the slice and the nearest obstacle be used as the potential function. The traces of the local maxima of the potential function, as the slice is swept through the work space or configuration space, are termed *freeways.*

The algorithm begins by tracing a path from the start and tracing a path from the goal onto the nearest freeways by gradient ascent on the potential function in the slices which intersect the start and goal positions. This corresponds to the accessibility and departability algorithms described in Section 2.3.5 and Section 2.3.6.

If the start and goal freeways are connected, then the algorithm terminates. In general, the set of freeways will not be connected, and paths between neighboring freeways must be found. The OPP freeways are connected via *bridge curves.* The bridge curves are constructed in the vicinity of *interesting critical points.* Interesting critical points occur when *channels* (Figure 1.5) join or split. Bridge curves are also built when freeways terminate in the free space at bifurcation points (where traces of

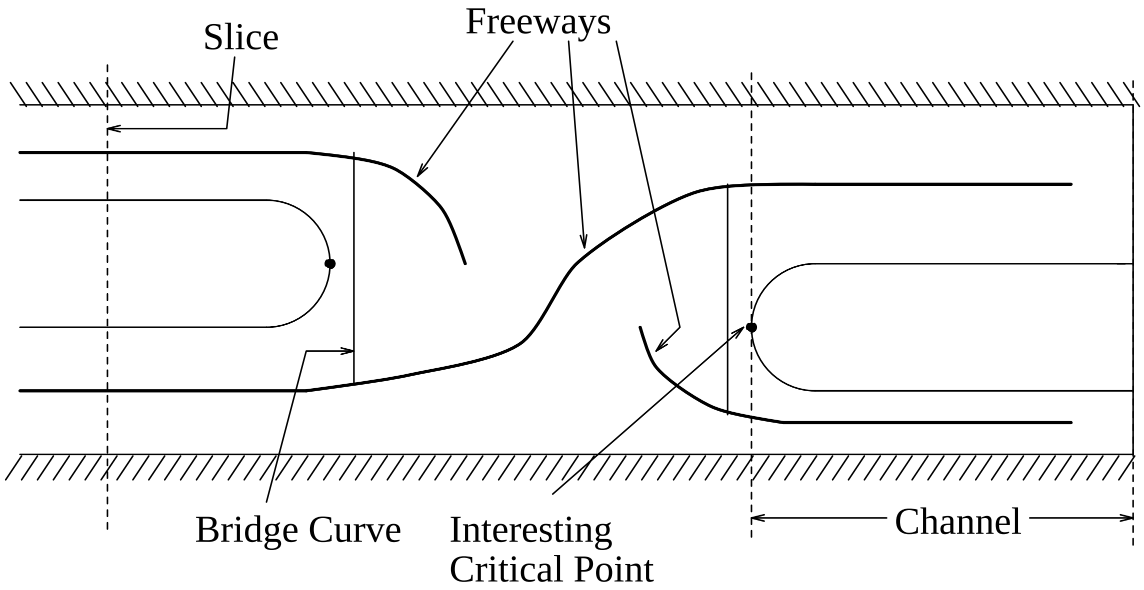

Figure 1.5 Schematic of the OPP planning scheme.

local maxima and local minima meet). A bridge curve is built leading away from a bifurcation point to another skeleton. This procedure is repeated until the start and goal freeway curves are connected, or all interesting critical points and bifurcation points have been explored in which case there does not exist a path between the start and the goal. The union of bridge and freeway curves, sometimes termed a *skeleton*, forms the one-dimensional roadmap.

Regarding sensor based planning, the OPP cannot be directly implemented in a sensor-based way because it assumes (1) prior knowledge of the location of all the interesting critical points; and (2) that freeway curves can be traced backward from the vantage point of the goal. One undesirable feature of this approach is that the freeways and location of interesting critical points are dependent upon sweep direction. In Section 4.7.1, it is shown that the freeway curves are a subset of the HGVG.

Cell Decomposition

In cellular decomposition (Chazelle, [10]; Latombe, [28]), robot's free space is decomposed into regions called *cells*. There are two types of cellular decomposition methods: exact and approximate. Exact cellular decomposition decomposes the free space into cells whose union is the free space. In this approach the shared boundary of adjacent cells typically has some sort of physical meaning such as a sudden change in the closest obstacle. In approximate cellular decomposition methods, the free space is decomposed into cells having pre-defined shapes like ike pixels in an image; typically, the union of all of the cells in approximate cellular decomposition is a subset of the entire free space. Motion planning is effected by searching a *connectivity graph*, a graph whose nodes are the cells, and whose edges correspond to adjacent cells.

Trapezoid Decomposition. An example of an exact cellular decomposition can be found in Figure 1.6. Each cell is either a triangle or a trapezoid and is formed by drawing a vertical line segment through each polygonal vertex. The end point of each segment is the boundary of the free space. Two cells are adjacent if they share a common boundary. The connectivity graph of the cellular decomposition of Figure 1.6 can be found in Figure 1.7. The connectivity graph can then be searched to find a path between the start and goal.

Generalized Voronoi Decomposition. This decomposition approach is related to the generalized Voronoi diagram discussed above. Figure Figure 1.8 contains an exact cellular decomposition in which each cell is the set of points closest to a particular object. The connectivity graph (Figure 1.9) of this environment is the "dual" of the generalized Voronoi diagram of the same environment (Figure 1.4). In this case, the connectivity graph is termed the *generalized Delaunay triangulation.* The generalized Voronoi diagram and the generalized Delaunay triangulation are duals in that one is connected if and only if the other is connected. This relationship will prove to be quite useful in demonstrating connectivity of roadmaps, such as the HGVG.

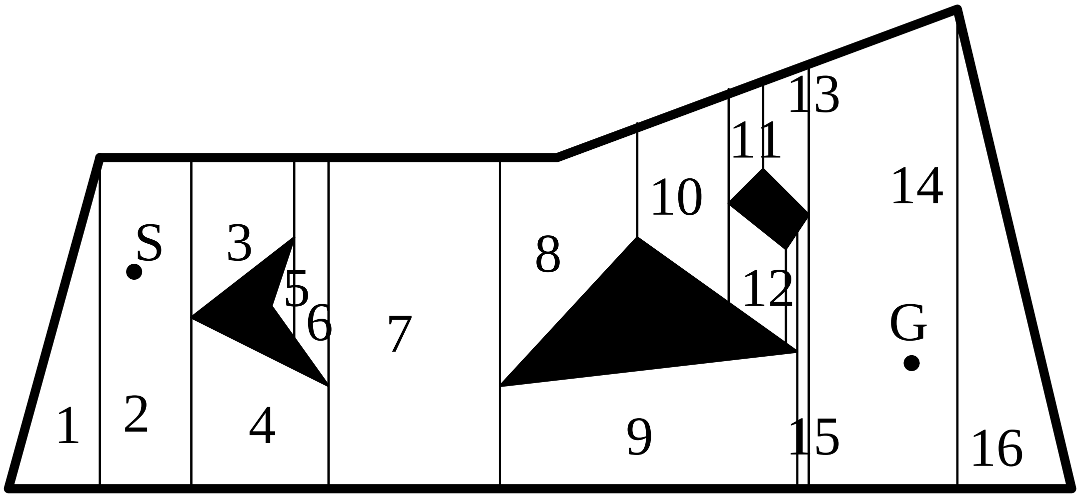

Figure 1.6 Bounded environment with three internal obstacles is decomposed into sixteen cells where the start location is in cell 2 and the goal location is in cell 14.

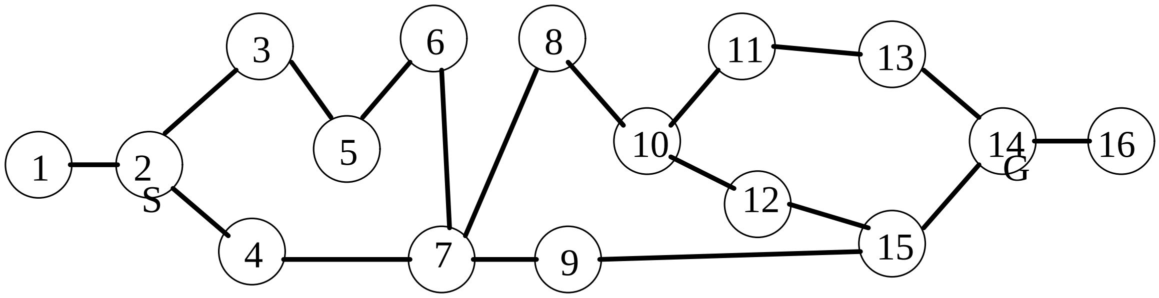

Figure 1.7 The connectivity graph of the cellular decomposition in Figure 1.6.

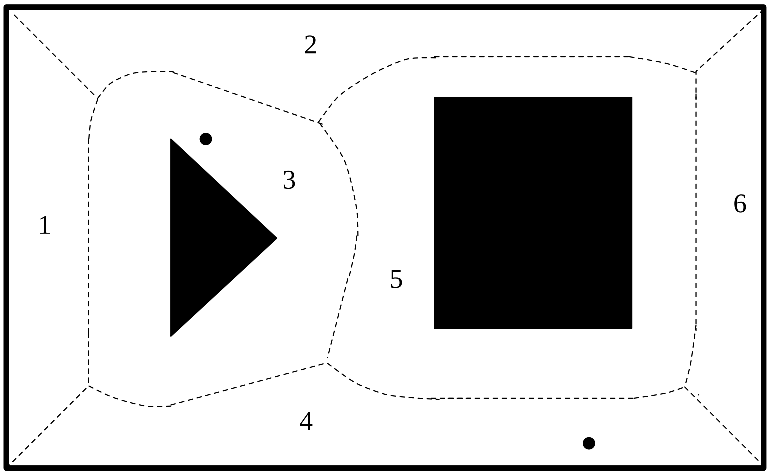

Figure 1.8 Exact cell decomposition where each cell is the set of points closest to a particular object.

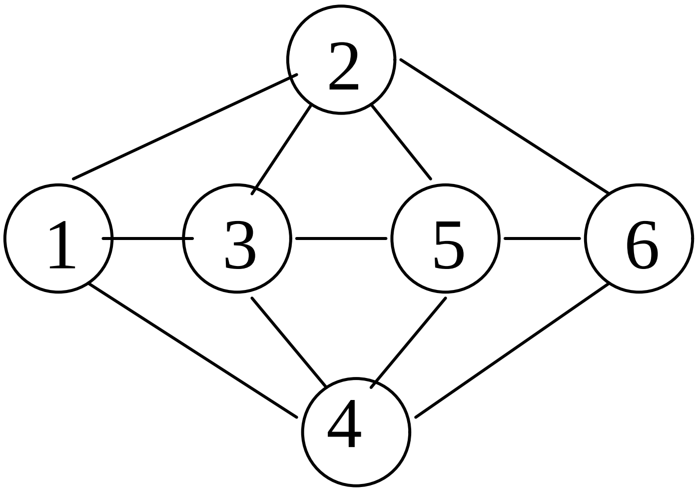

Figure 1.9 Connectivity graph for the cellular decomposition in Figure 1.8.

1.2.2 Prior Work in Sensor Based Planning

In contrast with classical robot motion planning, *sensor based* robot motion planning incorporates sensor information, reflecting the current state of the environment, into a robot's planning process. Sensor based planning has received increased attention lately, as it is requirement for realistic deployment of autonomous robots in unstructured environments. For a review of many sensor-based planning techniques, see Rao et al. [35].

Most current sensor based planning methods are heuristic algorithms which work very well under a variety of conditions, but there are no proofs of correctness that guarantee a path can be found. Moreover, there do not exist well established thresholds for when heuristic algorithms fail. Typically, these heuristic algorithms are limited to two-dimensional environments, where the geometry and topology of the problem is simple. One class of heuristic algorithms is a behavioral based approach in which the robot is armed with a simple set of behaviors such as following a wall [5]. A hierarchy of cooperating behaviors forms more complicated behaviors such as exploration. An extension of this type of approach is called sequencing [20, 21, 22]. Since there are strong experimental results indicating the utility of these approaches (such as [22]), some of these algorithms may provide a future basis for provably correct sensor based planners.

Another type of heuristic approach involves the discretization of a planar world into pixels (Borenstein and Koren [4]). In this approach, each pixel is assigned a value indicating the likelihood that it overlaps an obstacle. This method lends itself very nicely to implementation with real sensors, but discretizing the world may require a large amount of computer memory and may lead to an inaccurate representation of the world.

There are many non-heuristic algorithms for which provably correct solutions exist in the plane (see [35] for an overview). For example, Lumelsky's "bug" algorithm [31] is one of the first provably correct sensor based schemes to work in the plane. In this method, the robot travels along a straight line segment that connects the start and goal. If the robot encounters an obstacle, it turns left and begins to circumnavigate the obstacle. If the robot encounters the start-goal line segment at a point different from which it departed, the robot continues along the line segment towards the goal. Otherwise, the algorithm terminates, indicating that the robot cannot reach the goal.

Although the bug algorithm may seem simple, its contribution is that it provides a path to the goal or determines that it is *impossible* to ever reach the goal. However, this algorithm (like many described in [35]) requires knowledge of the goal's location during the planning process. Furthermore, this algorithm simply returns a path from the start to the goal. The resulting path does not reflect the topology of the free space (the region of the environment not occupied by obstacles) and thus, it cannot be used to guide future robot excursions.

Taylor and Kriegman's method [43] has a provably correct solution, uses realistic sensor assumptions, and need not require prior knowledge of the goal's location. In this method, the robot forms a graph of a bounded free space by circumnavigating each of the obstacles, and then creating an adjacency relationship between obstacles within line of sight of each other. However, this method requires landmarks in constructing its map and is limited to the planar case.

Another example of robotic exploration with realistic sensor assumptions can be found in Kortenkamp and Weymouth [27]. In this work, a map called a "topological map" of the world is formed. However, the algorithm in [27] assumes that the environment is populated with rectangular obstacles that meet at nearly ninety degree angles. Its application is therefore highly restricted.

One approach to sensor based motion planning is to adapt the structure of a provably correct classical motion planning scheme to a sensor based implementation. Roadmaps are one of the complete classical planning methods. An example of a complete roadmap scheme is Canny and Lin's Opportunistic Path Planner (OPP) [9]. Rimon adapted this motion planning scheme for sensor based use [38]. To our knowledge, this is the first sensor based motion planning scheme for workspaces whose dimension is greater than two. However, connectivity of the roadmap in [38] cannot be guaranteed without active perception. Furthermore, from a practical point of view, there are two limitations to Rimon's method. First, to construct the roadmap, the robot posseses an *interesting critical point sensor* and a *minimum passage sensor,* both of whose implementations are not well described. Second, a robust and detailed procedure for constructing the roadmap fragments from sensor data is not presented.

1.2.3 How the HGVG Relates to Previous Work

The HGVG is a roadmap that is an extension of the *generalized Voronoi diagram* (GVD) into higher dimensions. The GVD is the locus of points equidistant to two or more obstacles which are convex sets in the plane. (The Voronoi diagram (VD) is the set of points equidistant to two or more isolated points (sometimes termed sites) in the plane. See Aurenhammer [2] for survey on Voronoi diagrams.) The GVD was first used for motion planning in [40]. Active research in applying the GVD to motion planning began with Ó'Dúnlaing and Yap [34], who considered motion planning for a disk in the plane. However, the method in [34] requires full knowledge of the world's geometry prior to the planning event, and its retract methodology may not extend to non-planar problems. In [36], an incremental approach to create a Voronoi diagram-like structure, which is limited to the case of a plane, was introduced.

Prior work (e.g., Avis and Bhattacharya [3]) describes the *Voronoi graph*, an extension of the Voronoi diagram extended into higher dimensions. The Voronoi graph is the locus of points in m dimensions equidistant to m point sites. The *generalized Voronoi graph* (GVG), defined in Chapter 2, extends the Voronoi graph to the case of convex obstacles; that is, it is the set of points in m dimensions equidistant to m convex obstacles. Though the GVG introduced in this work appears to be new, a disconnected GVG-like structure for $SE(3)$ is described in [7].

The GVG, described in Chapter 2, can be thought of as the natural extension of the GVD into higher dimensions. However, unlike the GVD, the GVG is in general *not* a roadmap because it is not necessarily connected in dimensions greater than two. Therefore, in Chapter 3, we introduce additional structures, termed *higher order generalized Voronoi graphs* which are guaranteed to link the disconnected

GVG components into a connected network. The resulting connected structure is the *hierarchical generalized Voronoi graph* (HGVG).

The following diagram summarizes the evolution of the HGVG. The GVG and HGVG appear in bold because they are new structures introduced in this work.

	Point Sites		Convex Sets
Double Equidistance	VD	$\rightarrow$	GVD
			$\downarrow$
m-wise Equidistance	VG	$\rightarrow$	**GVG**
			$\downarrow$
m-wise Equidistance and Other Structures			**HGVG**

A key feature of the HGVG approach is that it is a roadmap which can be fully constructed using line of sight information, as described in Chapter 4. In fact, the original motivation of this work was to develop such techniques for Rimon's sensor based extension to the OPP method. However, in doing so, we developed the GVG, and then the HGVG representation which we believe is more amenable to sensor based implementation than the OPP-motivated approach. However, it should be noted that the HGVG incremental construction procedures in Chapter 4 are also applicable to the OPP method as well.

1.3 Contributions of Thesis

A major contribution of this thesis is the definition of the hierarchical generalized Voronoi graph (HGVG), a roadmap structure that is sufficient for motion planning in dimensions greater than or equal to two. The backbone of the HGVG is the generalized Voronoi graph (GVG) which is a natural extension of the generalized Voronoi diagram [34] into higher dimensions. Whereas the generalized Voronoi diagram is always one less dimension than the space in which it is defined, the GVG is always one-dimensional. The GVG, however, is not guaranteed to be connected, and is thus not directly applicable to motion planning. In order to guarantee connectivity, the GVG is augmented with additional structures, termed higher order generalized Voronoi graphs. The combined structure is the HGVG. The HGVG is a powerful motion planning tool because once the HGVG is constructed, it reduces motion planning in higher dimensional work or configuration spaces to a one-dimensional search problem.

We make no claim that the HGVG has any clear advantage over other methods for classical motion planning. Nevertheless, we demonstrate that it is sufficient for motion planning. In doing this, we introduce a structure, termed the generalized Voronoi complex, which can be used to show connectivity for a broad class of structures that includes the HGVG.

The second major contribution of this thesis is a well defined technique to incrementally construct the HGVG using only line of sight information. In general, there will not be a single vantage point from which the robot can "see" the entire world and therefore it must incrementally construct a roadmap as it explores its environment. The incremental construction procedure described in this thesis can also be used when full geometry of the world is available. Unlike other sensor based construction procedures, this procedure is proven to be complete (it is guaranteed to find the goal if it is reachable or to map the entire environment in finite time) and need not require any artificial landmarks nor abstract sensors. Additionally, this incremental construction technique in this thesis can be applied to other methods such as Canny and Lin's Opportunistic Path Planner, Rimon's extension, and wall following algorithms.

A key property of the GVG and the HGVG is that they are defined in terms of a distance function, $d_i(x)$, that is readily estimated by realistic sensors. The distance functions introduced in this thesis measure the distance to obstacles that are within line of sight of the robot.

Another contribution of this work is a method for determining extrema of the

Euclidean distance function which is nonsmooth at its local maxima and saddle points. This method makes use of only first order information without making use any type of nonsmooth second derivative.

Finally, we experimentally verify this approach for a mobile robot with a ring of sonar sensors. In doing this, we introduce a new approach to programming mobile robots. Instead of evaluating matrix and other arithmetic operations, the incremental construction technique is encoded in look-up tables. Thus an order one complexity operation replaces a complicated arithmetic operation. We also developed a simulator which generates the GVG in a three-dimensional Euclidean space.

1.4 Overview of Thesis

This thesis is organized into six chapters and three appendices, where the first six chapters are labeled Chapters 1, 2, 3, 4, 5, and 6, and four appendices are labeled Chapters A, B, and C. Chapter 1 introduces the reader to motion planning and sensor based motion planning in general, and motivates the goals of this work.

Chapter 2 defines the GVG and introduces some of its properties. In this chapter, some basic definitions and results of convex analysis, set theory, and topology are used. These definitions and results, with derivations, appear in Chapter A. Also in this chapter, the distance function is first defined and some of its properties are explored. More properties of the distance function are derived in Chapter B.

The GVG, in general, is not sufficient for motion planning, so it is augmented by additional structures. The GVG and the additional structures is the HGVG, which is defined in Chapter 3. Using results from Chapter C, the HGVG, with some additional links, is shown to be connected. This chapter also draws from the results in the appendices.

Chapter 4 describes the incremental construction technique for the HGVG. It also explicitly demonstrates how this technique can be applied to the freeways of the OPP method.

Finally, Chapter 5 describes the experimental implementation of this approach on an actual mobile robot, and Chapter 6 is the conclusion.

Chapter 2

The Generalized Voronoi Graph

2.1 Distance Functions

A function which encodes the distance between a robot and a nearby obstacle is key to our subsequent definitions and results. This section defines three types of distance function: the X-distance function, the S-distance function, and the V-distance function, each of which provides a rigorous geometric foundation for our definition of the roadmap. Since many realistic sensors readily measure distance between nearby obstacles and a robot, our roadmap structure lends itself nicely to sensor based implementation. A more complete discussion of these functions and their properties can be found in Section B.1.

2.1.1 X-Distance Function

Definition 2.1 (Single Object Distance Function) The distance between a point, x and a convex set C_i is

$$d_i(x) = \min_{c_0 \in C_i} \|x - c_0\|, \tag{2.1}$$

where $\|\cdot\|$ is the two-norm in $\mathbb{R}^m$.

In Clarke [19] it is shown that the gradient of $d_i(x)$ is

$$\nabla d_i(x) = \frac{x - c_0}{\|x - c_0\|}, \tag{2.2}$$

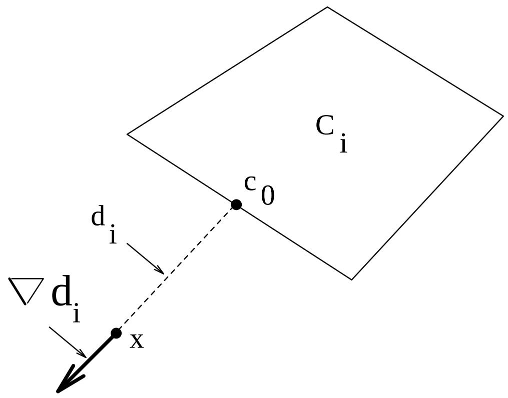

Figure 2.1 Distance between x and C_i is the distance to the closest point on C_i. The gradient is a unit vector pointing away from the nearest point.

where c_0 is the point closest to x in C_i. That is, c_0 is the point where $\|x - c_0\| = \min_{c \in C_i} \|x - c\|$. In later sections, we write $c_0 = \text{argmin } d_i(x)$. The gradient, $\nabla d_i(x)$, is a unit vector, based at x, pointing away from c_0 along a line defined by c_0 and x. For convex sets, the closest point is always unique. See Figure 2.1.

An important characteristic of $d_i(x)$ and $\nabla d_i(x)$ is that they can be *computed from sensor data.* For example, consider a mobile robot with a ring of sonar sensors (Figure 2.2). The sonar sensor measurement provides an approximate value of the distance function, and the direction opposite to which the sensor is facing approximates the distance gradient. That is, a sensor on the robot in Figure 2.2 points in the direction of the negated distance gradient.

Typically, the environment is populated with multiple obstacles, and thus we define

Definition 2.2 (Multi-Object Distance Function) The distance between a point x and the set of all obstacles, $C_1, \ldots, C_n$, in the environment is defined as

$$D(x) = \min_i d_i(x). \tag{2.3}$$

Accordingly, the distance between a point and the boundary of the free space is considered to be the distance to the nearest obstacle. This information can be

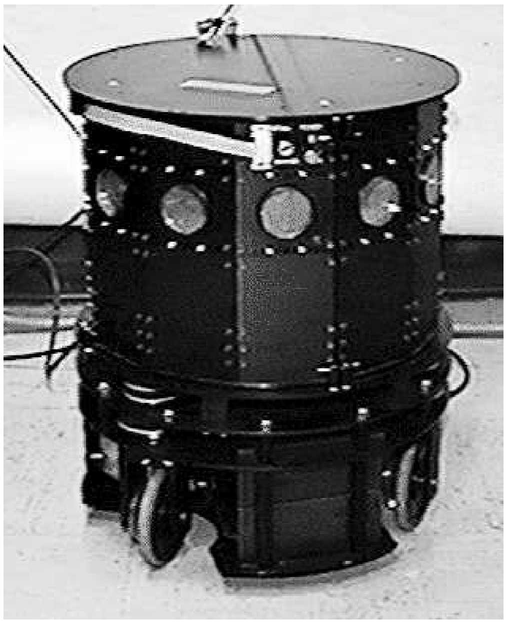

Figure 2.2 Mobile robot with sonar ring.

readily *computed from sensor information.* Note that the free space can be defined as $\mathcal{FS} = \{x \in \mathcal{W} : D(x) > 0\}$.

It is shown in Chapter B that the multi-object distance function is nonsmooth, and hence its gradient cannot be trivially defined. However, using nonsmooth analysis (which is reviewed in Section B.1), it can be shown that the *generalized gradient* of $D(x)$ is

$$\partial D(x) = \mathrm{Co}\{\nabla d_i(x) \, : \, i \in I(x)\}, \tag{2.4}$$

where: (1) Co is the convex hull operation, (2) ∂ is the generalized gradient operator, and (3) $I(x)$ is defined as the set of indices such that $\forall i \in I(x)$, each C_i is the closest object to x (x may be equidistant to two or more obstacles). See Figure 2.3. Notationally, if ∂ appears in front of a set, as opposed to a function, then it means the boundary of the set. Since $\partial D(x)$ is comprised of single object distance gradients, it can be *readily computed from sensor data.*

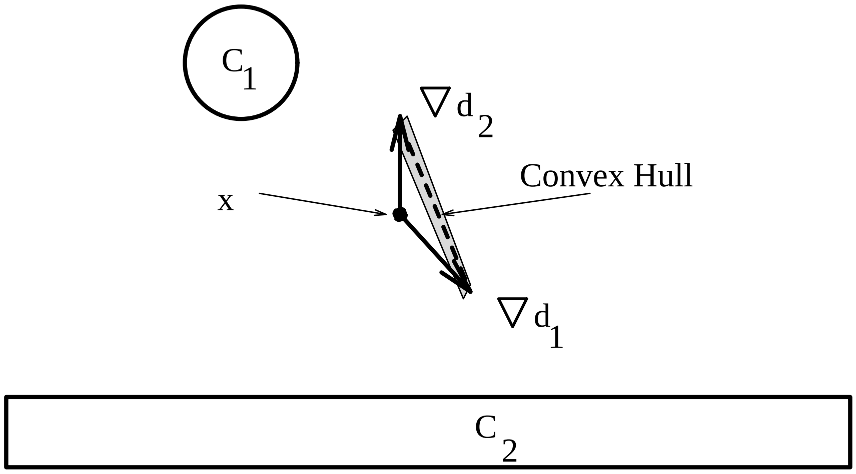

Figure 2.3 The generalized gradient of the multi-object distance function at a point is the convex hull of the gradients of the single object distance functions which correspond to the closest equidistant obstacles at that point. The solid arrows are single object distance gradients and the shaded region corresponds to the heads of all of the vectors which are in the convex hull of the two single object gradients.

In later sections, we will need distance measurements to the second closest obstacle, as well as the closest obstacle. Unfortunately, in some instances, the second closest obstacle may not be "within line of sight" at a given vantage point. Therefore, we must define another distance function which deals with occluded obstacles. For the sake of terminology, we will term the distance function defined in this section to be the *X-distance function* because its implementation assumes a robot can see through obstacles, as if the robot has X-ray vision.

2.1.2 The S-Distance Function

We want to develop a distance function that solely relies on line of sight measurements. With the X-distance function, the second closest obstacle may not be within line of sight of a point from which distance is being measured. In this section, we define a distance function that measures distance to an obstacle by computing the distance between a point, x, and the obstacle's nearest point that is within line of sight of x. If there are no points on the obstacle that are within line of sight of x, then the object is considered to be occluded and located at a distance of infinity. Since this distance function is based only upon line of sight information, it is more conducive to implementation with realistic sensors than the X-distance function.

A point c is *within line of sight* of x if there exists a straight line segment which connects x and c without penetrating any obstacle. That is, c is within line of sight of x if for all $t \in [0,1]$, $(x(1-t)+ct)$ lies in $\mathcal{FS}$.

Let $\tilde{C}_i(x)$ be the set of points on an object C_i that are within line of sight of x, i.e.,

$$\tilde{C}_i(x) = \{c \in C_i : x(1-t)+ct \in \mathcal{FS}, \forall t \in [0,1]\}.$$

If $\tilde{C}_i(x) \neq \emptyset$, then the obstacle, C_i, is *within slanted-line of sight* of x. Otherwise, if $\tilde{C}_i(x) = \emptyset$, then C_i is *occluded* or *fully occluded* at x. Note the difference between line of sight and slanted-line of sight: a point can be within line of sight and an obstacle can be within slanted-line of sight.

Definition 2.3 (Single Object S-Distance Function) The *S-distance function* measures the distance between a point x and the nearest point on an obstacle that is within slanted-line of sight of x. If no such line of sight point exists, then the distance is infinite, i.e.,

$$d_i^S(x) = \begin{cases} \min_{c \in \tilde{C}_i(x)} \|x-c\|, & \text{if } \tilde{C}_i(x) \neq \emptyset, \\ \infty, & \text{if } \tilde{C}_i(x) = \emptyset. \end{cases} \tag{2.5}$$

In order to delineate the difference between the two distance functions, we define d_i^X to be the X-distance function. The superscript is omitted when the distance function being considered is obvious. Figure 2.4 contrasts the two distance functions. In Figure 2.4(a), the shortest line of sight path from x to C_i is orthogonal to an edge of C_i, and thus $d_i^S(x) = d_i^X(x)$. However, in Figure 2.4(b), the shortest line of sight path from x to $C_i(x)$ is not orthogonal to an edge of C_i, and thus $d_i^S(x) \neq d_i^X(x)$. When the nearest feature on an obstacle is a co-dimension one face (e.g., an edge of a polygonal obstacle), as in this case, the shortest line of sight path is *slanted* and hence the S-distance function is also known as the *slanted distance function*.

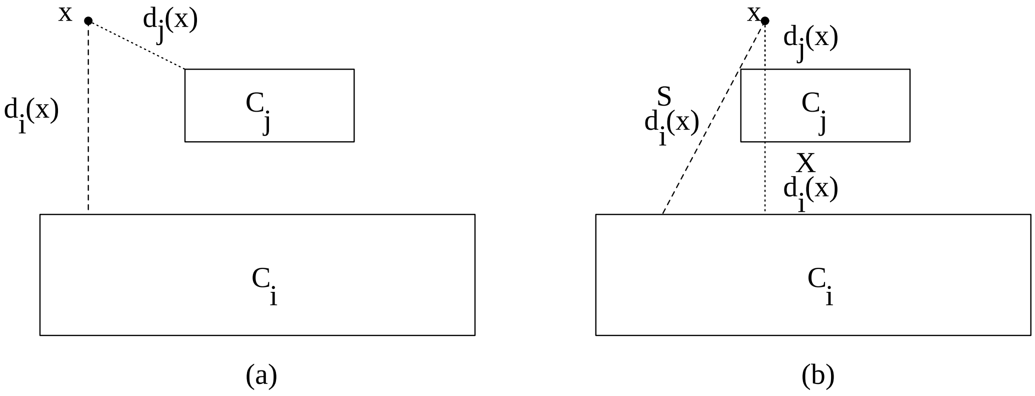

Figure 2.4 Comparison between S-distance and X-distance functions. In (a) the S-distance and X-distance functions are the same for both obstacles, but in (b) the S-distance and X-distance functions are the same only for object C_j.

Naturally, the multi-object slanted distance function D^S can be defined as the minimum over the single object slanted distance functions. It can be shown that $D^S(x) = D^X(x)$ for all points x in the free space.

Unfortunately, the gradient of d_i^S is not easily defined, as demonstrated in Figure 2.5. In this figure, the dashed lines represent the slanted distance from x to C_i. Recall that the gradient of a real valued function is the direction in which the function locally increases the most rapidly. The direction of the largest increase of distance is *not* necessarily in the direction away from the "closest" point, as it was with the X-distance function. The dotted lines represent the increased slanted distance to C_i by moving towards C_j. This distance is significantly greater than the distance obtained by moving in a direction away from the closest point. In this case, the slanted distance to C_i increases the most by approaching object C_j. Thus, the question of how best to define the S-distance function gradient is open at present.

2.1.3 The V-Distance Function

The S-distance function has the advantage of using solely line of sight information, but its ambiguous gradient definition makes it undesirable. In this section, we define a distance function, termed the *visible distance function*, which uses line of sight information and has a reasonable gradient definition.

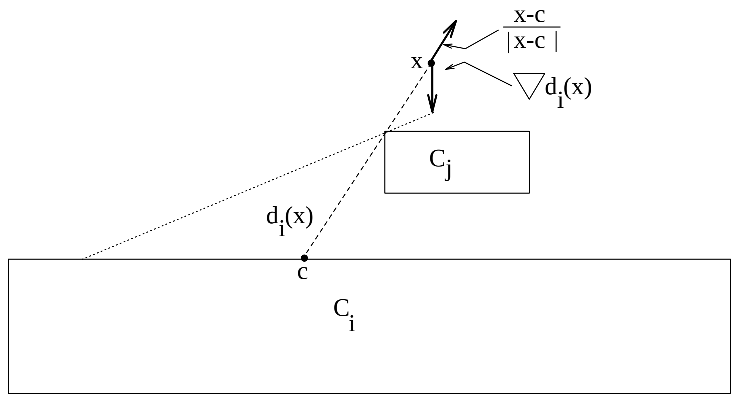

Figure 2.5 The vector, ∇d_i, the direction which maximally increases distance, is not $\frac{x-c}{|x-c|}$.

The definition of the V-distance function requires a different notion of line of sight. Let c be the nearest point in C_i to x, as defined by the X-distance function (i.e., $c = \text{argmin } d_i^X(x)$). The obstacle C_i is *within visible-line of sight* of a point x, if the line segment which connects c and x does not penetrate any other obstacle. In other words, C_i is within visible-line of sight of x if $c \in \tilde{C}_i(x)$, where we are using the same definition of $\tilde{C}_i(x)$ from Section 2.1.2. If $c \notin \tilde{C}_i(x)$, the obstacle is *visible-occluded.* The definition of *fully occluded* remains the same (if $\tilde{C}_i(x) = \emptyset$, then the obstacle is considered to be fully occluded from x). If $c \notin \tilde{C}_i(x)$ and $\tilde{C}_i(x) \neq \emptyset$, then C_i is *partially occluded.*

Definition 2.4 (Single Object V-Distance Function) The *V-distance function* measures the distance between a point x, and the nearest point that is within visible-line of sight of an obstacle. If the nearest point is not within visible-line of sight of x, then the distance is infinity, i.e.,

$$d_i^V(x) = \begin{cases} \min_{c \in C_i} \|x - c\|, & \text{if } c \in \text{int}(\tilde{C}_i(x)), \\ \infty, & \text{if } c \notin \text{int}(\tilde{C}_i(x)). \end{cases} \tag{2.6}$$

It is important to emphasize the difference between this definition and Definition 2.3. In Definition 2.3, the distance is finite if there exists a straight line path

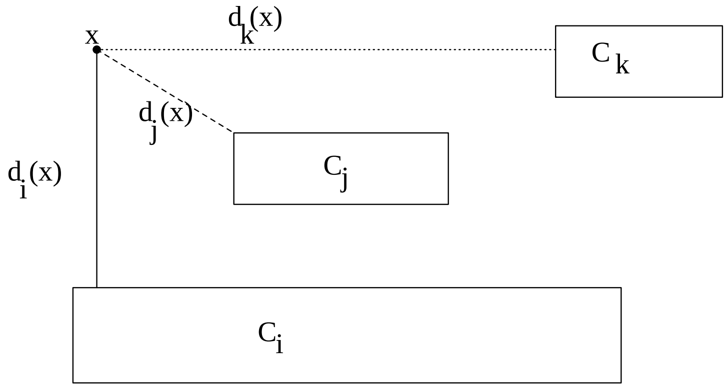

Figure 2.6 The corresponding values of X-distance function and V-distance function for each obstacle are the same.

between the point x and any point on the obstacle such that this path does not penetrate any other obstacle (i.e., $\tilde{C}_i(x) \neq \emptyset$). In Definition 2.4, the distance is finite if there exists a straight line path between x and the closest point on the obstacle, such that this path does not penetrate any other obstacle. (i.e., $c \in \tilde{C}_i(x)$). The V-distance function is referred to as the *visible distance function*.

In Figure 2.6, all of the obstacles are within visible-line (and slanted-line) of sight of x, but in Figure 2.7 with the *new* definition of line of sight to an obstacle, C_i is not within visible-line of sight of x. Thus, it is occluded, from x, making C_k the second closest obstacle. It should be noted that using the *new* definition of line of sight to an obstacle, although C_i is occluded, there do exist points on the boundary of C_i which are within line of sight of x. (Recall that points can be within line of sight and that obstacles can be within visible-line of sight.) That is, C_i is *not* fully occluded from x because $\tilde{C}_i(x) \neq \emptyset$.

Using visible-line of sight, the definition of occlusion is undesirable for sensor based implementation because there may exist scenarios for which sensor noise makes it difficult for the robot to infer whether the closest point to an object is within line of sight of the robot. In Figure 2.8(a), the closest point on obstacle C_i is within line of sight of the robot, and thus the distance is finite. However, in Figure 2.8(b), the closest point on obstacle C_i is occluded and thus the distance to

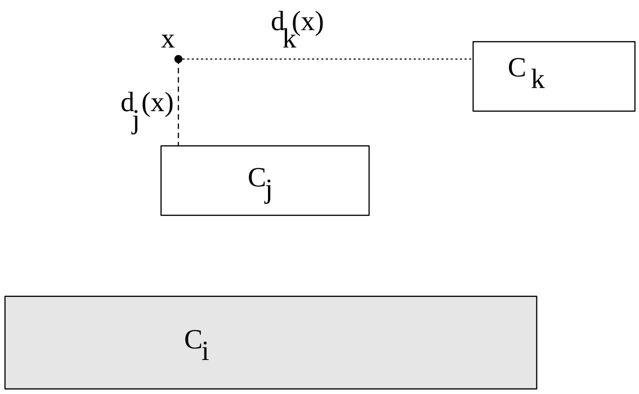

Figure 2.7 Using the V-distance function, the distance to C_i is infinity.

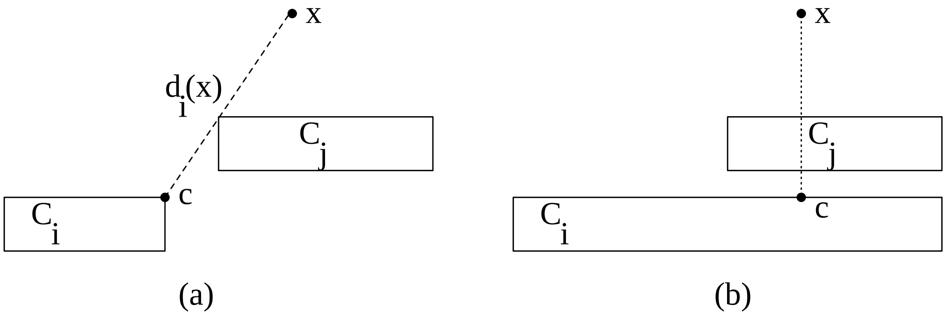

Figure 2.8 (a) The distance to obstacle C_i is finite because the dashed line does not touch (but is close) C_j. (b) The distance to obstacle C_i is infinite.

C_i is infinite. In both cases, the robot "sees" almost the same thing.

The advantage of the V-distance function is that its gradient is easy to define. Let $c = \operatorname{argmin} d_i(x)$. If $c \in \tilde{C}_i(x)$,

$$\nabla d_i^V(x) = \frac{x - c}{\|x - c\|}.$$

Naturally, there exists a multi-object visible distance function D^V which is defined as the minimum of the single object visible distance functions. It can be shown that $D^V(x) = D^S(x) = D^X(x)$ for all points x in the free space. In later sections, we omit the superscript on the distance function when it is either obvious or when

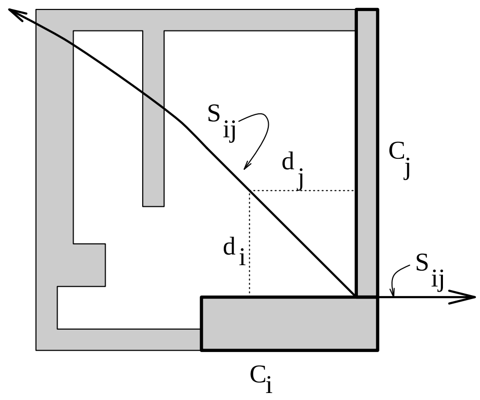

Figure 2.9 The solid line represents $\mathcal{S}_{ij}$, the set of points equidistant to obstacles C_i and C_j. Note that $\mathcal{S}_{ij}$ is unbounded and contains two components: the left component contains two linear sub-components and one parabolic sub-component, and the right component is linear. For all points x in the right component, $\nabla d_i(x) = \nabla d_j(x)$. The dotted lines emphasize that at a point on $\mathcal{S}_{ij}$, $d_i(x) = d_j(x)$.

it does not matter which distance function is being considered.

2.2 The Generalized Voronoi Graph

Using the previously defined distance functions, we can now define the generalized Voronoi diagram (GVD) and the generalized Voronoi graph (GVG). The GVD and GVG are explicitly defined in Section 2.2.1. Sections 2.2.2 and 2.2.3 contain the definitions of related structures which are useful in demonstrating some of the GVG's properties.

2.2.1 Equidistant Faces

The basic building block of the GVD and GVG is the set of points equidistant to two sets C_i and C_j, which we term the *two-equidistant surface*,

$$\mathcal{S}_{ij} = \{x \in \mathcal{W} \backslash (C_i \bigcup C_j) : d_i(x) - d_j(x) = 0\}. \tag{2.7}$$

See Figure 2.9. The superscripts are omitted because this surface can be defined in

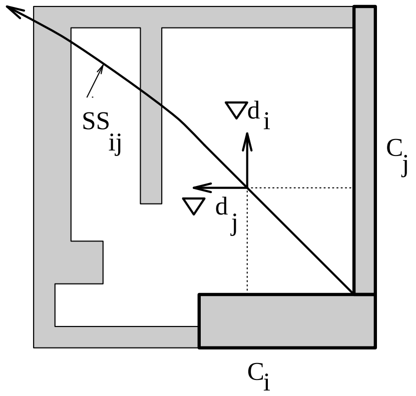

Figure 2.10 The solid line represents $\mathcal{SS}_{ij}$, the set of points equidistant to obstacles C_i and C_j such that the two closest points are distinct. Note, it is also unbounded and only has one component. Again, the dotted lines emphasize that for all points on $\mathcal{SS}_{ij}$, $d_i(x) = d_j(x)$ and the two vectors emphasize $\nabla d_i(x) \neq \nabla d_j(x)$.

terms of any of the distance functions defined in Section 2.1. Of particular interest is the subset of $\mathcal{S}_{ij}$ termed the *two-equidistant surjective surface*,

$$\mathcal{SS}_{ij} = \mathrm{cl}\{x \in \mathcal{S}_{ij} : \nabla d_i(x) \neq \nabla d_j(x)\}. \tag{2.8}$$

Note that In the interior of the free space, the single object distance function is smooth [19] because we are assuming our obstacles are convex.

The two-equidistant surjective surface, $\mathcal{SS}_{ij}$, is the set of points which are equidistant to two objects such that $\nabla d_i \neq \nabla d_j$, i.e., the function $\nabla(d_i - d_j)(x)$ is surjective for all $x \in \mathcal{SS}_{ij}$. This definition is required to deal with non-convex sets that are defined as the finite union of convex sets. See Figure 2.10. If $\mathcal{W}$ is solely populated with disjoint convex obstacles, then $\mathcal{SS}_{ij} = \mathcal{S}_{ij}$ for all i and j.

The union of all two-equidistant surfaces is called the *two-equidistant set*, $\mathbf{\mathcal{S}}^2$. Likewise, the union of all the two-equidistant surjective surfaces is called a *two-equidistant surjective set*, $\mathbf{\mathcal{SS}}^2$. That is,

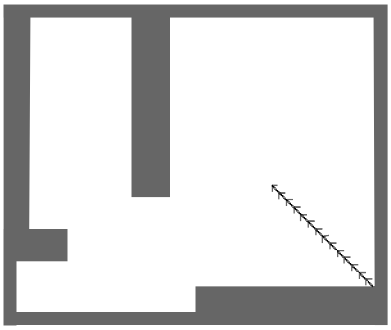

Figure 2.11 The solid line with angled ticks is the set of points equidistant and closest to obstacles C_i and C_j.

$$\mathcal{S}^2 = \bigcup_{i=1}^{n-1} \bigcup_{j=i+1}^{n} \mathcal{S}_{ij} \qquad \mathcal{SS}^2 = \bigcup_{i=1}^{n-1} \bigcup_{j=i+1}^{n} \mathcal{SS}_{ij}. \tag{2.9}$$

Definition 2.5 (Two-Equidistant Face) The *two-equidistant face*,

$$\mathcal{F}_{ij} = \{x \in \mathrm{cl}(\mathcal{SS}_{ij}) : d_i(x) = d_j(x) \leq d_h(x) \quad \forall h \neq i, j\}, \tag{2.10}$$

is the set of points equidistant to obstacles C_i and C_j, such that each point x in $\mathcal{F}_{ij}$ is closer to C_i and C_j than to any other obstacle.

By definition, $\mathcal{F}_{ij} \subset \mathrm{cl}(\mathcal{FS})$.

A two-equidistant face (Figure 2.11) is also termed a *generalized Voronoi face* in keeping with the conventions of the Voronoi diagram literature. Since the *closest two* obstacles define $\mathcal{F}_{ij}$, $d_i^X(x) = d_i^S(x) = d_i^V(x)$ for all points in $\mathcal{F}_{ij}$. Hence the definition of two-equidistant faces is independent of choice of distance function.

The *two-Voronoi set*, $\mathcal{F}^2$, is the union of all two-equidistant faces, i.e.,

$$\mathcal{F}^2 = \bigcup_{i=1}^{n-1} \bigcup_{j=i+1}^{n} \mathcal{F}_{ij}. \tag{2.11}$$

Since $\mathcal{F}^2$ is the set of points equidistant to two or more closest points on the boundary of a bounded space, it is the generalized Voronoi diagram (GVD) of that bounded

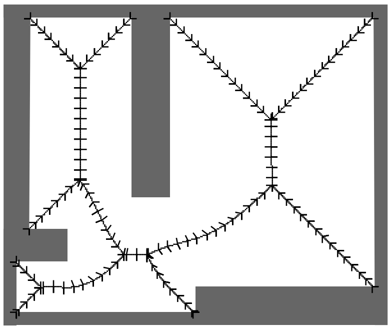

Figure 2.12 The ticked solid lines is the set of points equidistant to two obstacles, such that each edge fragment is closest to the equidistant obstacles.

space. See Figure 2.12.

To define the GVG, we continue to define lower dimensional subsets of $\mathcal{W}$. The *three-equidistant surface*, $\mathcal{S}_{ijk}$, is the set of points equidistant to three objects, C_i, C_j, and C_k. It can be defined in many ways:

$$\begin{aligned} \mathcal{S}_{ijk} &= \mathcal{S}_{ij} \bigcap \mathcal{S}_{jk} \bigcap \mathcal{S}_{ik}, \\ &= \mathcal{S}_{ij} \bigcap \mathcal{S}_{jk} = \mathcal{S}_{jk} \bigcap \mathcal{S}_{ik} = \mathcal{S}_{ij} \bigcap \mathcal{S}_{ik}, \\ &= \{x \in \mathcal{W} \backslash (C_i \bigcup C_j \bigcup C_k) : d_i(x) = d_j(x) = d_k(x)\}. \end{aligned} \tag{2.12}$$

Similarly, the *three-equidistant surjective surface*, $\mathcal{SS}_{ijk}$, a subset of $\mathcal{S}_{ijk}$, is the set of points equidistant to three objects, C_i, C_j, and C_k, such that for each point in $\mathcal{SS}_{ijk}$, the gradients of the individual single object distance functions are distinct. The three-equidistant surjective surface can be defined in many ways:

$$\begin{aligned} \mathcal{SS}_{ijk} &= \mathcal{SS}_{ij} \bigcap \mathcal{SS}_{jk} \bigcap \mathcal{SS}_{ik} \\ &= \mathcal{SS}_{ij} \bigcap \mathcal{SS}_{jk} = \mathcal{SS}_{jk} \bigcap \mathcal{SS}_{ik} = \mathcal{SS}_{ij} \bigcap \mathcal{SS}_{ik} \\ &= \{x \in \mathcal{W} \backslash (C_i \bigcup C_j \bigcup C_k) : d_i(x) = d_j(x) = d_k(x) \quad \text{such that} \\ &\qquad \nabla d_i(x) \neq \nabla d_j(x), \\ &\qquad \nabla d_i(x) \neq \nabla d_k(x), \text{ and} \\ &\qquad \nabla d_j(x) \neq \nabla d_k(x)\}. \end{aligned} \tag{2.13}$$

The union of all three-equidistant surfaces is called the *three-equidistant set*, $\mathbf{\mathcal{S}}^3$. Likewise, the union of all three-equidistant surjective surfaces is called the *three-equidistant surjective set*, $\mathbf{\mathcal{SS}}^3$. Consequently,

$$\begin{aligned} \mathbf{\mathcal{S}}^3 &= \bigcup_{i=1}^{n-2} \bigcup_{j=i+1}^{n-1} \bigcup_{k=j+1}^{n} \mathcal{S}_{ijk}, \\ \mathbf{\mathcal{SS}}^3 &= \bigcup_{i=1}^{n-2} \bigcup_{j=i+1}^{n-1} \bigcup_{k=i+1}^{n} \mathcal{SS}_{ijk}. \end{aligned} \tag{2.14}$$

The *three-equidistant face*, $\mathcal{F}_{ijk}$, a subset of $\mathcal{SS}_{ijk}$, is the set of points equidistant to C_i, C_j, and C_k, such that each point is closer to C_i, C_j, and C_k, than to any other object. That is, a three-equidistant face has the property that for all $x \in \mathcal{F}_{ijk}$ and for all $h \neq i, j, k$, $d_h(x) \geq d_i(x) = d_j(x) = d_k(x)$. Similarly, the *three-Voronoi set* is the union of all the three-equidistant faces. In other words,

$$\begin{aligned} \mathcal{F}_{ijk} &= \mathcal{F}_{ij} \bigcap \mathcal{F}_{ik} \bigcap \mathcal{F}_{jk}, \\ &= \mathcal{F}_{ij} \bigcap \mathcal{F}_{ik} = \mathcal{F}_{ij} \bigcap \mathcal{F}_{jk} = \mathcal{F}_{ik} \bigcap \mathcal{F}_{jk}, \\ \mathbf{\mathcal{F}}^3 &= \bigcup_{i=1}^{n-2} \bigcup_{j=i+1}^{n-1} \bigcup_{k=j+1}^{n} \mathcal{F}_{ijk}. \end{aligned} \tag{2.15}$$

Continuing in this vein, after taking the appropriate $k-2$ intersections, one can define a *k-equidistant surface*, $\mathcal{S}_{i_1 \ldots i_k}$, and a *k-equidistant surjective surface*, $\mathcal{SS}_{i_1 \ldots i_k}$. The Boundedness Assumption (Assumption 1.1) guarantees that there exists "enough" obstacles such that $\mathcal{S}_{i_1 \ldots i_k}$ and $\mathcal{SS}_{i_1 \ldots i_k}$ are not empty. After taking the appropriate unions, one can form the *k-equidistant set*, $\mathbf{\mathcal{S}}^k$, and the *k-equidistant surjective set*, $\mathbf{\mathcal{SS}}^k$. A subset of $\mathbf{\mathcal{SS}}^k$ of particular interest is the *k-equidistant face*, $\mathcal{F}_{i_1 \ldots i_k}$, which is the set of points equidistant to objects $C_{i_1}, \ldots, C_{i_k}$ such that each point is closer to objects $C_{i_1}, \ldots, C_{i_k}$ than to any other object. The *k-Voronoi set*, $\mathbf{\mathcal{F}}^k$, is simply the union of all k-equidistant faces:

$$\begin{aligned} \mathcal{F}_{i_1 \ldots i_k} &= \mathcal{F}_{i_1 i_2} \bigcap \mathcal{F}_{i_1 i_3} \bigcap \cdots \bigcap \mathcal{F}_{i_1 i_k}, \\ &= \mathcal{F}_{i_1 i_2 \ldots i_{k-1}} \bigcap \mathcal{F}_{i_1 i_k}, \\ \mathbf{\mathcal{F}}^k &= \bigcup_{i_1=1}^{n-k+1} \bigcup_{i_2=i_1+1}^{n-k+2} \cdots \bigcup_{i_k=i_{k-1}+1}^{n} \mathcal{F}_{i_1 i_2 \ldots i_k}. \end{aligned} \tag{2.16}$$

In m dimensions, the *generalized Voronoi edge*, and a *generalized Voronoi vertex* are respectively an m-equidistant face, $\mathcal{F}_{i_1 \ldots i_m}$, and $m+1$-equidistant face, $\mathcal{F}_{i_1 \ldots i_{m+1}}$. In Section 2.3.2, it is shown that a generalized Voronoi edge is one-dimensional, while a generalized Voronoi vertex is a point where generalized Voronoi edges meet. A generalized Voronoi vertex is typically termed a *meet point.* Using these definitions, we can define:

Definition 2.6 (Generalized Voronoi Graph) The *generalized Voronoi graph* (GVG) is defined as the collection of generalized Voronoi edges and vertices:

$$GVG = (\mathfrak{F}^m, \mathfrak{F}^{m+1}). \tag{2.17}$$

Each GVG edge is the set of points equidistant to m objects, such that each point is closer to these m objects than to any other object. An important characteristic of the GVG is that it is defined in terms of a distance function, which can be *computed readily from sensor data.* It is this feature that makes the GVG useful for sensor based motion planning.

Example 2.7 Figure 2.13 depicts a GVG in $\mathbb{R}^3$ for a rectangular enclosure with dimensions a, b, and c. The GVG edges, delineated by solid lines, constitute the locus of points equidistant to three obstacles, and the meet points are where the GVG edges intersect. In this example, there are twelve GVG edges: four of them outline a rectangle which represents the two-equidistant face defined by the top and bottom faces of the enclosure, and the remaining eight are spokes which terminate on the boundary.

Let the lower left corner of the box be the origin of a right handed coordinate system. Let the floor be the set of points $\{x \in \mathcal{W} : x^1 \in [0, a],\ x^2 \in [0, b],\ \text{and } x^3 = 0\}$, where x^i is the ith coordinate of a point x. Similarly, let the ceiling be the set of points $\{x \in \mathcal{W} : x^1 \in [0, a],\ x^2 \in [0, b],\ \text{and } x^3 = c\}$. The set of points equidistant to these two surfaces is $\{x \in \mathcal{W} : x^3 = \frac{c}{2}\}$. This is the two-equidistant surface and since for each point in the two-equidistant surface, the gradients are distinct and thus, this surface is also the two-equidistant surjective surface.

Now consider the front face and floor. The front face is the set of points $\{x \in \mathcal{W} : x^1 \in [0,a],\, x^2 = 0 \text{ and } x^3 \in [0,c]\}$. A two equidistant surface, defined by these obstacles, is $\{x \in \mathcal{W} : x^1 = x^3\}$. However, only for points, where x^1 and x^3 are positive, the distance gradients are distinct. Therefore, the two-equidistant surjective surface is $\{x \in \mathcal{W} : x^2 = x^3 \text{ and } x^3 > 0\}$. This two-equidistant surjective surface intersects the two-equidistant surjective surface defined by the floor and ceiling, at a line such that this line separates the floor/ceiling two-equidistant surjective surface into two regions: one where points are closer to the ceiling and floor than to the front face, and one where points are closer to the front face than to the ceiling and floor. This line ($\{x \in \mathcal{W} : x^2 = x^3 = \frac{c}{2}\}$) is the three-equidistant surjective surface defined by the floor, ceiling, and front face.

Three more lines, $\{x \in \mathcal{W} : x^1 = a - \frac{c}{2},\, x^3 = \frac{c}{2}\}$, $\{x \in \mathcal{W} : x^2 = b - \frac{c}{2},\, x^3 = \frac{c}{2}\}$, and $\{x \in \mathcal{W} : x^1 = \frac{c}{2},\, x^3 = \frac{c}{2}\}$ respectively exist for the right, back, and left faces of the rectangular enclosure. On the two-equidistant surjective surface, all four lines enclose a region where points are closer to the floor and ceiling than to any other obstacle. This region is the two-equidistant face defined by the floor and ceiling and are the set of points $\{x \in \mathcal{W} : x^1 \in [\frac{c}{2}, a - \frac{c}{2}], x^2 \in [\frac{c}{2}, b - \frac{c}{2}], x^3 = \frac{c}{2}\}$.

Since the floor and ceiling's two-equidistant face "hovers" above the floor at a height of $\frac{c}{2}$, the four GVG edges that outline this face have a height of $\frac{c}{2}$ with respect to the floor. In fact, each point on these GVG edges must be a distance $\frac{c}{2}$ from the corresponding obstacles that define the appropriate edge. The length of front and back edge is $a - c$, and the length of the left and right edges is $b - c$.

The four meet points associated with the floor and ceiling's two-equidistant face have coordinates, $(\frac{c}{2}, \frac{c}{2}, \frac{c}{2})^T$, $(a - \frac{c}{2}, \frac{c}{2}, \frac{c}{2})^T$, $(a - \frac{c}{2}, b - \frac{c}{2}, \frac{c}{2})^T$, and $(\frac{c}{2}, b - \frac{c}{2}, \frac{c}{2})^T$, in counter clock-wise order starting from the lower left meet point. Out of each meet point, emanates two edges that terminate on the boundary.

A useful method for visualizing the GVG is to picture a sphere rolling along the side of the room, touching three obstacles. The center of this sphere traces a GVG edge. This imaginary sphere touches four obstacles when its center coincides with a meet point. The corner graph components can be visualized by an expanding sphere

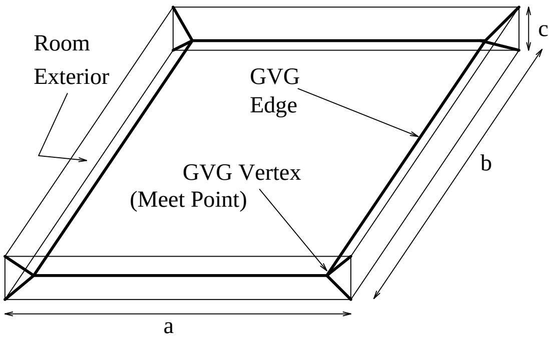

Figure 2.13 The generalized Voronoi graph in a rectangular enclosure. The solid lines represent the GVG edges, which meet at GVG vertices, sometimes called meet points.

that maintains contact with three walls. ♦

2.2.2 Generalized Voronoi Region

For subsequent analysis, it is useful to define the following.

Definition 2.8 The Generalized Voronoi Region, $\mathfrak{F}_i$, *is the closure of the set of points closer to one particular obstacle than to any other obstacle. In other words,*

$$\mathfrak{F}_i = \{x \in \mathcal{FS} : \, d_i(x) \leq d_h(x) \;\; \forall h \neq i\}. \tag{2.18}$$

It can be shown that the $\mathfrak{F}_i$ is *generalized star shaped*; this means that for all x in $\mathfrak{F}_i$, there is a closest point c_i in C_i to x such that the segment $\overline{xc_i}$ is fully contained in $\mathfrak{F}_i$. If C_i were a point, then $\mathfrak{F}_i$ would be star shaped.

2.2.3 Boundary Face and Floating Boundary Face

It is shown in the following section that the $k+1$-equidistant faces are on the boundaries of the k-equidistant faces. However, the $k+1$-equidistant faces are not the only geometric structures which can exist on the boundaries of the k-equidistant faces. The *k-boundary face* and the *floating k-boundary face*, defined below, are also on the boundary of k-equidistant faces. It is shown below that $k+1$-equidistant faces, k-boundary faces and floating k-boundary faces are the only structures on

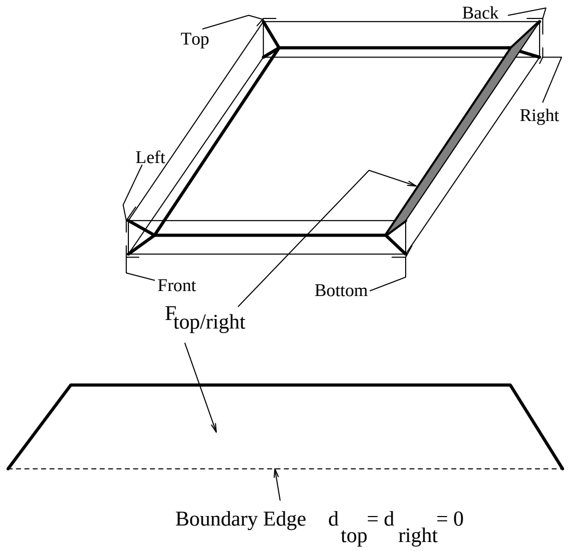

Figure 2.14 An example of a two-equidistant face that contains a boundary face as a portion of its boundary. The boundary face is represented by light dotted lines whereas the generalized Voronoi edges are represented by dark solid lines.

the boundaries of the k-equidistant faces. These structures are essential to the connectivity of our roadmap, and in incremental tracing routines of the GVG.

Definition 2.9 (k-Boundary Face) The set of points on the boundary of the free space where k obstacles intersect is the *k-boundary face* and is defined as

$$C_{i_1 \ldots i_k} = \{x \in \mathcal{F}_{i_1 \ldots i_k} \text{ such that } \; D(x) = 0\}. \tag{2.19}$$

In m dimensions, an $(m-1)$-boundary face is termed a *boundary edge*, as illustrated in Figure 2.14. Notationally, we represent the intersection of two obstacles C_i and C_j as C_{ij}. *Boundary fragments* are connected subsets of the boundary edges, and are denoted c_{ij} $(c_{ij} \subset C_{ij})$.

The *floating k-boundary surface*, $FS_{i_1 \ldots i_k}$, is the set of points on the boundary of a k-equidistant surjective surface where two gradient vectors become equal.

$$FS_{i_1 \ldots i_k} = \{x \in \mathcal{SS}_{i_1 \ldots i_k} : \; \nabla d_{j_1}(x) = \nabla d_{j_2}(x) \text{ where } j_1, j_2 \in \{i_1, \ldots, i_k\}\}\,. \tag{2.20}$$

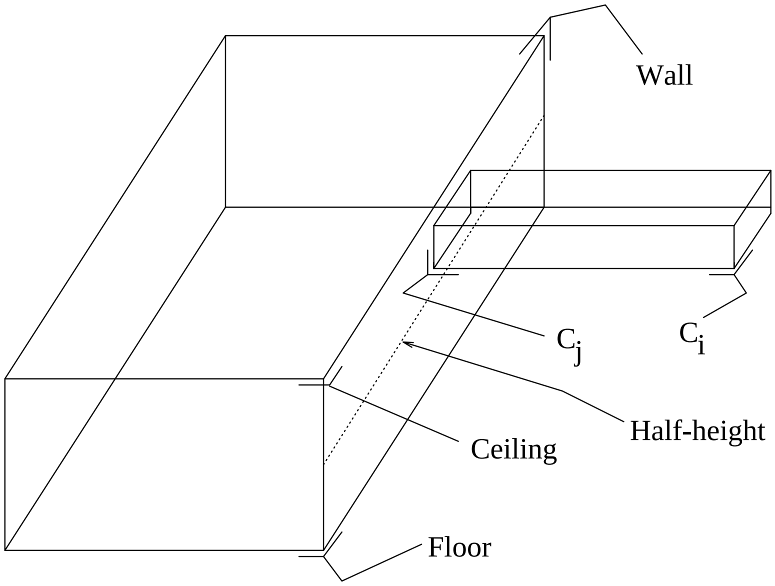

Figure 2.15 A room with a hole in its side wall. The dotted line marks the half-height of the room.

The only other boundary component of a k-equidistant face is called a *floating k-boundary face*, $FC_{i_1 \ldots i_k}$. It is the set of points in a floating k-boundary surface where each point in the set is closest to the k closest obstacles. $FC_{i_1 \ldots i_k}$ is the boundary of a k-equidistant face where two or more gradient vectors become collinear, and it is denoted

$$FC_{i_1 \ldots i_k} = \{x \in FS_{i_1 \ldots i_k} : \forall h \notin \{i_1, \ldots, i_k\},\, d_h(x) \geq d_{i_1}(x) = \cdots = d_{i_k}(x)\}\,. \tag{2.21}$$

Analogous to boundary edges, a *floating boundary edge* is a floating $(m-1)$-boundary face in $\mathbb{R}^m$. Furthermore, *boundary fragments* are connected subsets of the boundary edges, and are denoted by fc_{ij} where $fc_{ij} \subset FC_{ij}$.

Example 2.10 (Hole) The room depicted in Figure 2.15 contains a hole or duct in one of its side walls. The bottom and front faces of the duct are labeled C_i and C_j, respectively. Note also that the duct enters the room at a height higher than half the distance between the floor and ceiling.

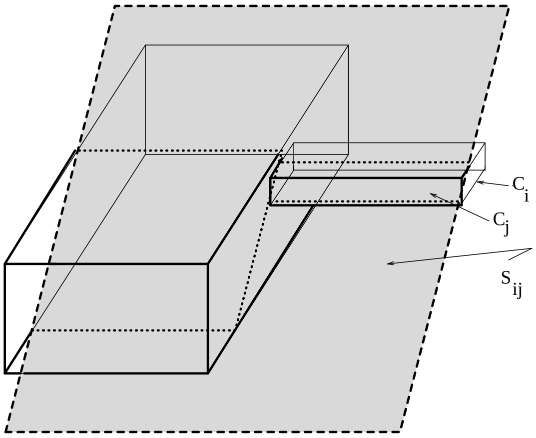

Figure 2.16 The shaded region is $\mathcal{S}_{ij}$. It is bordered by thick dashed lines to emphasize that $\mathcal{S}_{ij}$ is unbounded. The thick dotted lines correspond to points where $\mathcal{S}_{ij}$ intersects the rectangular enclosures. The thick solid lines on the rectangular enclosures are drawn to emphasize the features on the enclosure that are in front of $\mathcal{S}_{ij}$ in this view.

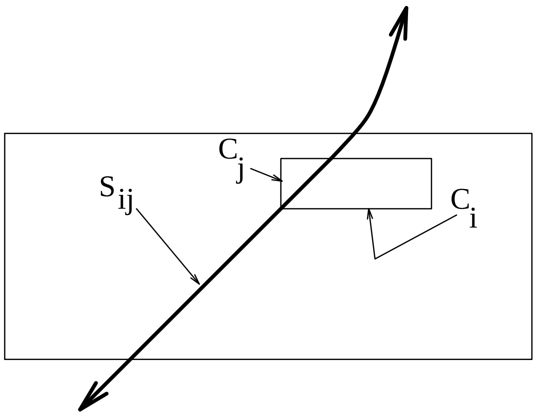

Figure 2.17 Side view of Figure 2.16.

Consider the two-equidistant face defined by the incoming duct's front and bottom faces, C_i and C_j. The shaded region in Figure 2.16 is a portion of a two-equidistant surface, $\mathcal{S}_{ij}$, defined by objects C_i and C_j. The boundary of the shaded region is dotted to emphasize that it is unbounded (i.e., a sphere of finite radius cannot contain the two-equidistant surface). Figure 2.17 illustrates a side view of Figure 2.16. Here, the two-equidistant surface $\mathcal{S}_{ij}$ is represented as a solid line with arrows on both sides to emphasize that it is unbounded.

Recall that a two-equidistant surjective surface is the set of points in a two-equidistant surface such that the gradients to each defining obstacle do not coincide. The shaded region in Figure 2.18 is a portion of the two-equidistant surjective surface, $\mathcal{SS}_{ij}$, defined by C_i and C_j. Note, this region is unbounded. Figure 2.19 illustrates the side view of Figure 2.18 where the solid line represents the unbounded $\mathcal{SS}_{ij}$. The floating boundary surface (solid line Figure 2.18) is the portion of bound-

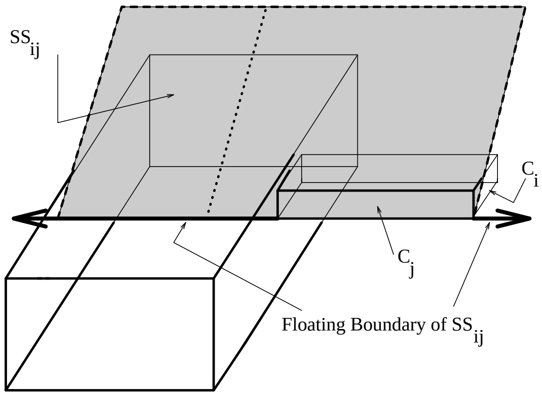

Figure 2.18 The shaded region $\mathcal{SS}_{ij}$ which is also unbounded. The thick double arrowed line represents the boundary of $\mathcal{SS}_{ij}$, which is also unbounded. The dotted line simply represents a path in $\mathcal{SS}_{ij}$.

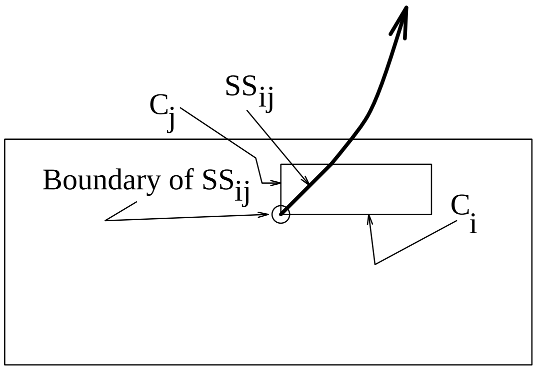

Figure 2.19 $\mathcal{SS}_{ij}$ (Side view).

ary of the two-equidistant surjective surface where the two distance function gradient vectors coincide (Figure 2.20).

The shaded region in Figure 2.21 depicts the two-equidistant face, $\mathcal{F}_{ij}$, which is bounded. The boundary of $\mathcal{F}_{ij}$ consists of three parts: a GVG edge, a boundary edge, and a floating boundary edge. The boundary edge runs along the intersection of the boundaries of obstacles C_i and C_j. $D(x)$ is zero for all points on this portion of $\mathcal{F}_{ij}$. The floating boundary edge is the set of points in $\mathcal{F}_{ij}$ where $D(x)$ is greater than zero, but $\nabla d_i(x) = \nabla d_j(x)$. The nearest point to all points on the floating boundary edge is a corner of the duct entrance. This corner is the lower left corner of the duct entrance in Figure 2.22, which is a side view of Figure 2.21. ♦

2.3 Basic Properties of the GVG

We now prove two important properties of the GVG: (1) it is one-dimensional, and (2) it is accessible from any point in $\mathcal{FS}$.

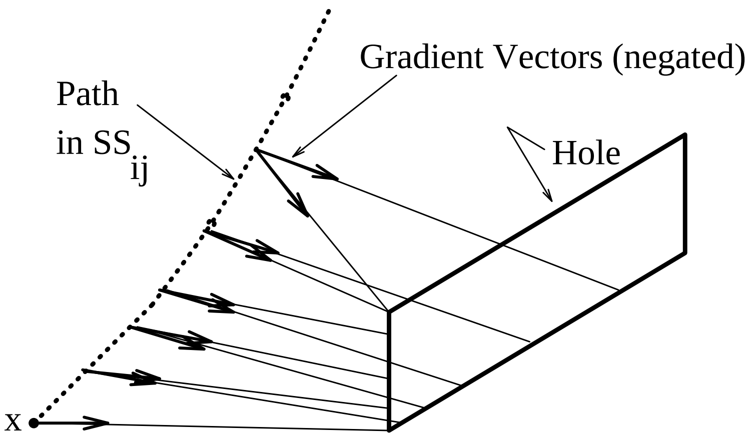

Figure 2.20 The dotted line represents a path in $\mathcal{SS}_{ij}$ depicted in Figures 2.18 and 2.19. At each point, there is a pair of arrows which are unit vectors pointing to the closest points on the closest objects. These unit vectors are the negated distance function gradients. The thin lines connect a point on the path to the closest points on the closest obstacles. As points in the path approach x, the unit vectors become collinear. This point is in the floating boundary edge.

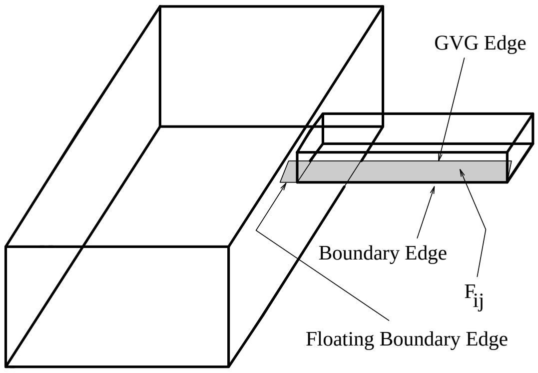

Figure 2.21 $\mathcal{F}_{ij}$.

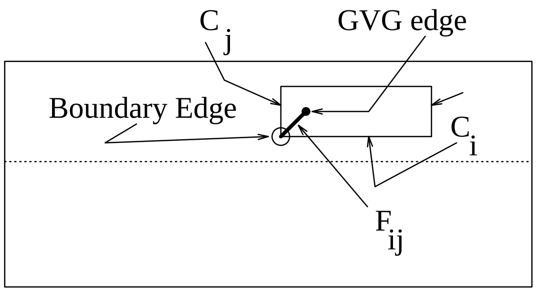

Figure 2.22 $\mathcal{F}_{ij}$ (Side view).

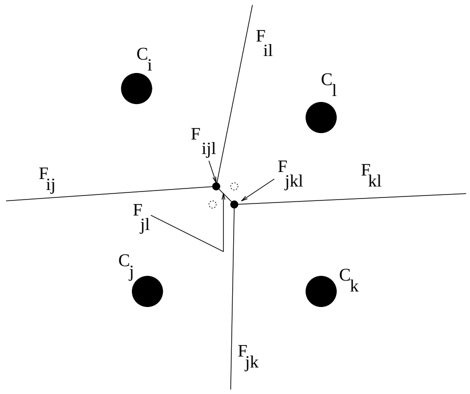

Figure 2.23 Even though $\mathcal{F}_{ij}$ exists, $\mathcal{F}_{ijk}$ does not.

2.3.1 GVG Boundary Elements

The following results are needed to show that the GVG is composed of one-dimensional pieces. These results are also important in proving accessibility and connectivity of the GVG.

Proposition 2.11 *If the $(k+1)$-equidistant face $\mathcal{F}_{i_1 \ldots i_{k+1}}$ is non-empty, then the k-equidistant face $\mathcal{F}_{i_1 \ldots i_k}$ must also be non-empty; however, the converse is not necessarily true. Furthermore,*

$$\partial \mathcal{F}_{i_1 \ldots i_k} = \mathcal{F}_{i_1 \ldots i_k i_{k+1}} \bigcup C_{i_1 \ldots i_k} \bigcup FC_{i_1 \ldots i_k}.$$

Proof: The first part of the proof is a consequence of the definition of the $(k+1)$-equidistant face, $\mathcal{F}_{i_1 \ldots i_k i_{k+1}}$, which is $\mathcal{F}_{i_1 \ldots i_k} \bigcap \mathcal{F}_{i_1 i_{k+1}}$. By definition, if $\mathcal{F}_{i_1 \ldots i_k} = \emptyset$ then $\mathcal{F}_{i_1 \ldots i_{k+1}} = \emptyset$. The contrapositive is also true, if $\mathcal{F}_{i_1 \ldots i_{k+1}} \neq \emptyset$ then $\mathcal{F}_{i_1 \ldots i_k} \neq \emptyset$. Note that the converse is not necessarily true. In Figure 2.23, there exists an edge, $\mathcal{F}_{ij}$, but no vertex, $\mathcal{F}_{ijk}$.

The second part of this proof follows from Proposition A.10 and Lemma A.11 (Section A.1).

Lemma A.11 *For two subsets A and B of $\mathbb{R}^m$, the closure of the intersection of A and B is a subset of the intersection of the closures of A and B. In other words,*

$$\mathrm{cl}(A \bigcap B) \subset \mathrm{cl}(A) \bigcap \mathrm{cl}(B).$$

Proposition A.10 *For two subsets A and B of $\mathbb{R}^m$, the boundary of the intersection of the two sets obeys the following inclusion:*

$$\partial(A \bigcap B) \subset \left(\partial A \bigcap \mathrm{cl}(B)\right) \bigcup \left(\partial B \bigcap \mathrm{cl}(A)\right).$$

Let A, B, and C be the sets with the following definitions:

$$\begin{aligned}
A &= \{x \in \mathcal{W} : \forall h,\, d_h(x) \geq d_{i_1}(x) = \cdots = d_{i_k}(x)\}, \\
B &= \{x \in \mathcal{W} : \forall p, q \in \{i_1, \ldots, i_k\}\,, \nabla d_p(x) \neq \nabla d_q(x)\}\,, \\
C &= \{x \in \mathcal{W} : D(x) > 0\}\,.
\end{aligned}$$

It can easily be seen that $\mathcal{F}_{i_1 \ldots i_k} = A \bigcap \mathrm{cl}(B \bigcap C)$. By Proposition A.10, Lemma A.11, and the relationship $\partial\, \mathrm{cl}\,(B \bigcap C) = \partial\,(B \bigcap C)$, it follows that

$$\begin{aligned}
\partial \mathcal{F}_{i_1 \ldots i_k} &= \partial\,(A \bigcap \mathrm{cl}\,(B \bigcap C)) \\
&\subset [\partial A \bigcap \mathrm{cl}\,(\mathrm{cl}\,(B \bigcap C))] \quad \bigcup [\partial\, \mathrm{cl}\,(B \bigcap C) \bigcap \mathrm{cl}\,(A)] \\
&= [\partial A \bigcap \mathrm{cl}\,(B \bigcap C)] \quad \bigcup [\partial\,(B \bigcap C) \bigcap \mathrm{cl}\,(A)] \\
&\subset [\partial A \bigcap \mathrm{cl}(B) \bigcap \mathrm{cl}(C)] \quad \bigcup [([\partial B \bigcap \mathrm{cl}\,(C)] \bigcup [\partial C \bigcap \mathrm{cl}\,(B)]) \bigcap \mathrm{cl}(A)] \\
&= [\partial A \bigcap \mathrm{cl}(B) \bigcap \mathrm{cl}(C)] \quad \bigcup [\partial B \bigcap \mathrm{cl}(A) \bigcap \mathrm{cl}(C)] \quad \bigcup [\partial C \bigcap \mathrm{cl}(A) \bigcap \mathrm{cl}(B)] \\
&= \mathcal{F}_{i_1 \ldots i_{k+1}} \quad \bigcup FC_{i_1 \ldots i_k} \quad \bigcup C_{i_1 \ldots i_k}
\end{aligned}$$

∎

2.3.2 Dimension of GVG Components

To determine the generic dimension of the GVG edges, we will use the pre-image theorem below to show that the intersection of a k-equidistant face and a two-equidistant face is $(k-1)$-dimensional. In order to properly invoke the pre-image theorem to obtain a correct dimension count, we first introduce an important transversality assumption and discuss its implications.

Assumption 2.12 (The Equidistant Surface Transversality Assumption) If equidistant surjective surfaces are manifolds, then they intersect transversally. That is, $\mathcal{SS}_{i_1\ldots i_k j_1} \pitchfork \mathcal{SS}_{i_1\ldots i_k j_2}$ with respect to $\mathcal{SS}_{i_1\ldots i_k}$ if $j_1 \neq j_2$.

In the case that $m = 2$ and the obstacles are points, this assumption is equivalent to the "no four points are co-circular" assumption which is often made in the Voronoi diagram literature. Assumption 2.12 is the generalization of this statement. This transversality assumption can also be interpreted as an assumption on the stability of the equidistant surface intersection geometry. In the left diagram of Figure 2.24, $\mathcal{SS}_{ijk} = \mathcal{SS}_{jkl} = \mathcal{SS}_{ikl} = \mathcal{SS}_{ijl}$ because there exists a circle which intersects the four obstacles (a non-generic case). After a slight perturbation of the obstacles, the equidistant surfaces no longer coincide (Figure 2.24). Since $\mathcal{SS}_{ijk}$ and $\mathcal{SS}_{ijl}$ are points in this example, they intersect transversally only if they do not intersect at all. As a result of Assumption 2.12, $\mathcal{SS}_{i_1\ldots i_k j_1} \neq \mathcal{SS}_{i_1\ldots i_k j_2}$ if and if only $j_1 \neq j_2$. The condition where two equidistant surjective surfaces are equal is an unstable non-generic one, and thus we do not consider it because any slight perturbation of the obstacle locations drastically affects equidistance relationships. In Figure 2.24, $\mathcal{SS}_{ijk} = \mathcal{SS}_{jkl} = \mathcal{SS}_{ikl} = \mathcal{SS}_{ijl}$, but after a slight perturbation of C_i, all the equidistant faces are no longer coincidental and $\mathcal{SS}_{ijkl} = \emptyset$.

We now show that the GVG is truly one-dimensional. This is done by iteratively invoking the pre-image theorem. Recall that for two manifolds M and N of dimension m and n, respectively, the pre-image theorem (also known as the submersion theorem) [1] is:

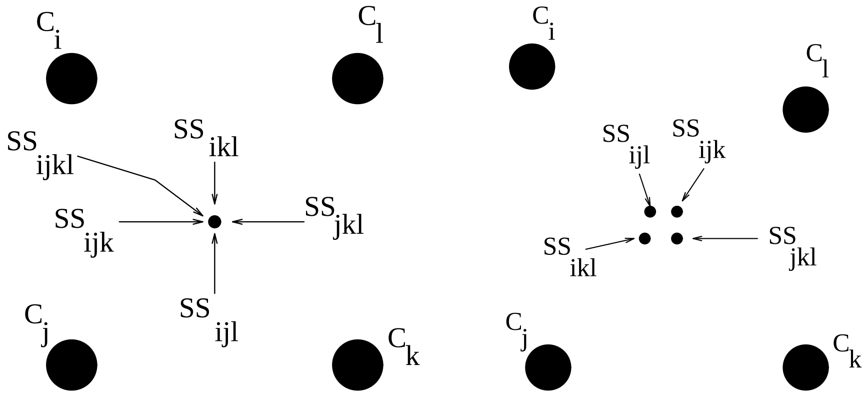

Figure 2.24 Left: non-generic arrangement. Right: small perturbation in obstacle locations.

Theorem 2.13 (Pre-image Theorem) *Let $f\colon M \to N$ be a C^∞ function and $n \in N$ be a regular value of f, then $f^{-1}(n) = \{m : m \in M,\, f(m) = n\}$ is a closed sub-manifold of M with tangent space given by $T_m f^{-1}(n) = \ker(T_m f)$. If N is finite dimensional,* $\operatorname{codim}(f^{-1}(n)) = \dim(N)$*, i.e.,* $\dim(f^{-1}(n)) = \dim(M) - \dim(N)$.

Recall that for a convex obstacle C_i, $d_i\colon M \to \mathbb{R}$ is a smooth function in the free space. When the regularity conditions are satisfied, the pre-image of the distance function has one less dimension than M, i.e., suppose $d_i(x) = r$ for $x \in M$ and $r \in \mathbb{R}$, the $\dim(d^{-1}(r)) = \dim(M) - \dim(\mathbb{R}) = \dim(M) - 1$.

To show that the edges are one-dimensional, we invoke the pre-image theorem $m-1$ times on the difference of two distance functions.

Lemma 2.14 *A two-equidistant surjective surface, $\mathcal{SS}_{ij}$, is a co-dimension manifold. $\mathcal{SS}^2$ is the union of co-dimension one manifolds.*

Proof: First, note that the function $(d_i - d_j)\colon \mathcal{W}\backslash(C_i \bigcup C_j) \to \mathbb{R}$ is smooth [19] because C_i and C_j are convex. Recall that the two-equidistant surjective surface $\mathcal{SS}_{ij}$ is a subset of $\mathcal{S}_{ij}$ such that $\nabla d_i(x) \neq \nabla d_j(x)$, for all $x \in \mathcal{SS}_{ij}$. This implies that $\nabla(d_i - d_j)(x)$ is surjective, and thus zero is a regular value of $(d_i - d_j)$, for all points $x \in \mathcal{SS}_{ij}$. By the pre-image theorem, $\mathcal{SS}_{ij}$ is a manifold having co-dimension one in $\mathcal{W}\backslash(C_i \bigcup C_j)$. $\mathbb{SS}^2$ is not a manifold, but by definion is the finite union of co-dimension one manifolds. ■

Note that at this point, we cannot infer that $\mathcal{F}_{ij}$ also has co-dimension one just because $\mathcal{F}_{ij}$ is a subset of $\mathcal{SS}_{ij}$.

Lemma 2.15 *A three-equidistant surjective surface, $\mathcal{SS}_{ijk}$, has co-dimension two in $\mathbb{R}^m$. $\mathbb{SS}^3$ is the union of co-dimension two manifolds.*

Proof: The three-equidistant surjective surface $\mathcal{SS}_{ijk}$ can be defined as $\mathcal{SS}_{ijk} = \{x \in \mathcal{SS}_{ij} : d_i(x) - d_k(x) = 0\}$. By Assumption 2.12, $\mathcal{SS}_{ij} \neq \mathcal{SS}_{ik}$ if and only if $i \neq k$, and thus, zero is a regular value of $(d_i - d_k)(x)$ on $\mathcal{SS}_{ij}$ for all $x \in \mathcal{SS}_{ijk}$. By the pre-image theorem, $\mathcal{SS}_{ijk}$ has co-dimension one in $\mathcal{SS}_{ij}\backslash C_k$, and thus co-dimension two in $\mathcal{W}\backslash(C_i \bigcup C_j \bigcup C_k)$. By definition $\mathbb{SS}^3$ is the finite union of co-dimension two manifolds. ■

By induction, one can show that the set of points equidistant to k obstacles has co-dimension one in the set of points equidistant to $k-1$ objects, and therefore this set has co-dimension $k-1$ in $\mathbb{R}^m$. That is, a k-equidistant surjective surface, $\mathcal{SS}_{i_1 \ldots i_k}$, has co-dimension $(k-1)$ in $\mathcal{W}\backslash \bigcup_{h=1}^{k} C_h$, or is $(m-k+1)$-dimensional. Hence, $\mathbb{SS}^m$ comprises one-dimensional components in $\mathbb{R}^m$, and $\mathbb{SS}^{m+1}$ is a set of points in $\mathbb{R}^m$.

By a similar argument like above, k-boundary faces are $(m-k+1)$-dimensional. For the moment, assume the obstacle boundaries are smooth (note that there exists a small enough ϵ such that for all C_i, $d_i^{-1}(\epsilon)$ forms a set of smooth "bloated" convex obstacles whose resulting equidistant faces have the same topology as the original set of equidistant faces defined by C_i). Now, when the set $\{x \in \partial C_{i_1} \bigcap \cdots \partial C_{i_k} \bigcap \mathrm{cl}(\mathcal{FS}) : \nabla d_{i_p}(x) \neq \nabla d_{i_q}(x) \, \forall p, q \in \{1, \ldots, k\}$ where $p \neq q\}$ is not empty, the pre-image theorem asserts that the pre-image of zero under the map $(d_{i_1}, \cdots, d_{i_k})^T\colon\ \mathrm{cl}(\mathcal{FS}) \to \mathbb{R}^k$ is $(m-k)$-dimensional.

The following lemma provides us with the necessary tool to demonstrate that the GVG edges are one-dimensional manifolds with zero-dimensional boundary. It is not sufficient to say that $\mathcal{F}_{i_1 \dots i_k}$ is $(m-k+1)$-dimensional because it is a subset of $\mathcal{SS}_{i_1 \dots i_k}$.

The following lemma draws upon some basic topology definitions. A *topology* ([1], p. 2) is a set X together with a collection O of subsets called open sets such that

1. $\emptyset \in O$,
2. if $O_1, O_2 \in O$, then $O_1 \bigcap O_2 \in O$, and
3. the union of any collection of open sets is open.

If A is a subset of the topological space X with topology O, the *subspace topology* ([1], p. 18) on A is defined by $O_A = \{O_i \bigcap A : O_i \in O\}$. It should be noted that open sets of the subspace topology may or may not be open sets in the original topology.

Lemma 2.16 *The interior of the k-equidistant face,* $\mathrm{int}(\mathcal{F}_{i_1 \dots i_k})$, *has the same dimension as the k-equidistant surjective surface,* $\mathcal{SS}_{i_1 \dots i_k}$.

Proof: The interior of the k-equidistant face, $\mathrm{int}(\mathcal{F}_{i_1 \dots i_k})$, has the property that for all $x \in \mathcal{FS}$, $d_{i_1}(x) = \cdots = d_{i_k}(x) < d_h(x) \quad \forall h \notin \{i_1, \dots, i_k\}$. Let $x \in \mathrm{int}(\mathcal{F}_{i_1 \dots i_k})$. Therefore, $x \in \mathcal{SS}_{i_1 \dots i_k}$. At a point x where $d_{i_1}(x) = \cdots = d_{i_k}(x) < d_h(x)$, there exists a nbhd$(x)$ in $\mathcal{SS}_{i_1 \dots i_k}$. Let $Y = \mathrm{nbhd}(x) \bigcap \mathcal{SS}_{i_1 \dots i_k}$. In order to show that $\mathrm{int}(\mathcal{F}_{i_1 \dots i_k})$ has the same dimension as $\mathcal{SS}_{i_1 \dots i_k}$, it suffices to show that $Y \subset \mathrm{int}(\mathcal{F}_{i_1 \dots i_k})$.

Since $x \in \mathrm{int}(\mathcal{F}_{i_1 \dots i_k})$, there exists an $h \notin \{i_1, \dots, i_k\}$ such that $d_{i_1}(x) = \cdots = d_{i_k}(x) < d_h(x)\}$. By continuity of the single object distance function, for ϵ sufficiently small, $d_{i_1}(x+\epsilon) = \cdots = d_{i_k}(x+\epsilon) < d_h(x+\epsilon)$. Therefore, Y is an open subset of $\mathrm{int}(\mathcal{F}_{i_1 \dots i_k})$, and thus the dimension of $\mathcal{SS}_{i_1 \dots i_k}$ and $\mathrm{int}(\mathcal{F}_{i_1 \dots i_k})$ are the same. ∎

Hence, by Lemma 2.16, $\mathrm{int}(\mathfrak{F}^m)$ and $\mathrm{int}(\mathfrak{F}^{m+1})$ are one and zero-dimensional, respectively.

Proposition 2.17 *The GVG edges are one-dimensional in $\mathcal{FS}$, and the GVG vertices are points.*

Proof: Since $\mathcal{SS}_{i_1 \ldots i_{m+1}}$ is a zero-dimensional manifold, it consists of isolated points in $\mathcal{W} \backslash \bigcup_{h=1}^{m} C_h$. Each of these points is an open set in the topology that $\mathcal{W} \backslash \bigcup_{h=1}^{m} C_h$ induces on $\mathcal{SS}_{i_1 \ldots i_{m+1}}$. Furthermore, non-empty subsets of a collection of points are points, and thus all non-empty subsets of $\mathcal{SS}_{i_1 \ldots i_{m+1}}$ are open sets in the subspace topology. Since $\mathcal{F}_{i_1 \ldots i_{m+1}}$ is a non-empty subset of $\mathcal{SS}_{i_1 \ldots i_{m+1}}$, $\mathcal{F}_{i_1 \ldots i_{m+1}}$ is zero-dimensional. A similar argument exists for $C_{i_1 \ldots i_m}$ and $FC_{i_1 \ldots i_m}$.

By Proposition 2.11, $\mathcal{F}_{i_1 \ldots i_m}$ can be defined as

$$\mathrm{int}(\mathcal{F}_{i_1 \ldots i_m}) \bigcup_{\beta} \mathcal{F}_{i_1 \ldots i_m k_\beta} \bigcup C_{i_1 \ldots i_m} \bigcup FC_{i_1 \ldots \imath_m}.$$

Since $\forall \beta$, $\mathcal{F}_{i_1 \ldots i_m k_\beta} \bigcup C_{i_1 \ldots i_m} \bigcup FC_{i_1 \ldots i_m}$ is zero-dimensional and $\mathrm{int}(\mathcal{F}_{i_1 \ldots i_m})$ is one-dimensional, the generalized Voronoi edge, $\mathcal{F}_{i_1 \ldots i_m}$, is a one-dimensional manifold with a zero-dimensional boundary. ■

The procedure described in the above paragraph can be repeated to show that any k-equidistant face is $(m-k+1)$-dimensional.

2.3.3 Complexity of the GVG

The following is useful in determining the complexity of the GVG.

Proposition 2.18 *Given Assumption 2.12 in $\mathbb{R}^m$, there exist $m+1$ generalized Voronoi edges emanating from a generalized Voronoi vertex.*

Proof: By Proposition 2.11, if $\mathcal{F}_{i_1 \ldots i_{m+1}}$ is not empty, then $\mathcal{F}_{j_1 \ldots j_m}$ is not empty where $\{j_1, \ldots j_m\}$ are m distinct elements from $\{i_1 \ldots i_{m+1}\}$. The number of m combinations of $m+1$ elements is $m+1$ ($\binom{m+1}{m} = m+1$). ■

This result has been previously determined for the special case where each object is a point site [26].

2.3.4 The Multi-object Distance Function is Smooth on a k-Equidistant Face

The multi-object distance function is nominally not smooth. However, when constrained to a k-equidistant surjective surface (and thus a k-equidistant face), the multi-object distance function is indeed smooth. This result is based upon the following two lemmas which are also used in the HGVG incremental construction procedure in Chapter 4.

Let the *regular Voronoi graph (RVG)* denote the Voronoi graph for the case in which the obstacles are points. The *regular k-equidistant face*, $\mathcal{R}_{i_1 \ldots i_k}$, is a k-equidistant face whose k closest obstacles are points. In m dimensions, a regular m-equidistant face is an RVG edge and it is equidistant to m closest point objects. Such a structure has been previously termed a Voronoi graph edge [3], but here, we will label it a *regular* Voronoi edge to better distinguish it from a generalized Voronoi edge.

Let x be a point on a k-equidistant face, and let $\{c_i\}$ denote the set of closest points of the k closest obstacles, $\{C_i\}$, to x. The regular k-equidistant face defined by the points $\{c_i\}$ and the k-equidistant face coincide at x. We can compute many items of interest about the GVG by exploiting this coincidence.

Lemma 2.19 *Let $c_1, \ldots, c_k$ be the k closest obstacle points to a point x. Let $\mathcal{R}_{i_1 \ldots i_k}$ be the regular Voronoi graph edge defined by these points where $k \leq m$. When Assumption 2.12 is upheld, any vector in the tangent space $T_x\mathcal{R}_{i_1 \ldots i_k}$ is orthogonal to the $k-1$-dimensional plane containing $c_1, \ldots, c_k$. The tangent space $T_x\mathcal{R}_{i_1 \ldots i_k}$ is also orthogonal to the $(k-1)$-dimensional plane containing the heads of the gradient vectors based at x.*

Proof: It can be shown that given Assumption 2.12, the *regular* k-equidistant faces (k-equidistant faces of the RVG) are diffeomorphic to $\mathbb{R}^{m-k+1}$ and thus a regular k-equidistant face is isometric to its tangent space. In other words, $T_x\mathcal{R}_{i_1 \ldots i_k} \simeq \mathcal{R}_{i_1 \ldots i_k}$. Assumption 2.12 guarantees that there exists a unique $(k-1)$-dimensional plane which contains $\{c_1, \ldots, c_k\}$, termed the *base plane*. The base plane is the affine hull of $\{c_1, \ldots, c_k\}$. Assumption 2.12 also guarantees that embedded in the base plane, there is a unique $(k-2)$-dimensional sphere, S, defined by $\{c_1, \ldots, c_k\}$. Finally, there exists a cone whose vertex is x and whose base is S (Figure 2.25). Every point on the centerline of this cone is always equidistant to $\{c_1, \ldots, c_k\}$, and therefore is contained in $\mathcal{R}_{i_1 \ldots i_k}$. Since $T_x\mathcal{R}_{i_1 \ldots i_k} \simeq \mathcal{R}_{i_1 \ldots i_k}$, this centerline is also contained in $T_x\mathcal{R}_{i_1 \ldots i_k}$.

Define an m-dimensional coordinate system whose origin is the center of the sphere S. The line that connects the origin of this coordinate system and x is the centerline. In this coordinate system, $x \in \mathcal{R}_{i_1 \ldots i_k}$ has coordinates $(x^1, \ldots, x^{m-k+1}, 0, \ldots, 0)^T$. All points on $s \in S$ have coordinates $(0, \ldots, 0, s^{m-k+2}, \ldots, s^m)^T$. Furthermore, the base plane can be represented by $\lambda(0, \ldots, 0, s^{m-k+2}, \ldots, s^m)^T$ for all $\lambda \in \mathbb{R}\backslash\{0\}$. Since

$$\langle (x^1, \ldots, x^{m-k+1}, 0, \ldots, 0)^T, (0, \ldots, 0, s^{m-k+2}, \ldots, s^m)^T \rangle = 0,$$

the base plane is orthogonal to the centerline. Therefore, $T_x\mathcal{R}_{i_1 \ldots i_k}$ is perpendicular to the base of the cone, which is the plane that contains the k closest points.

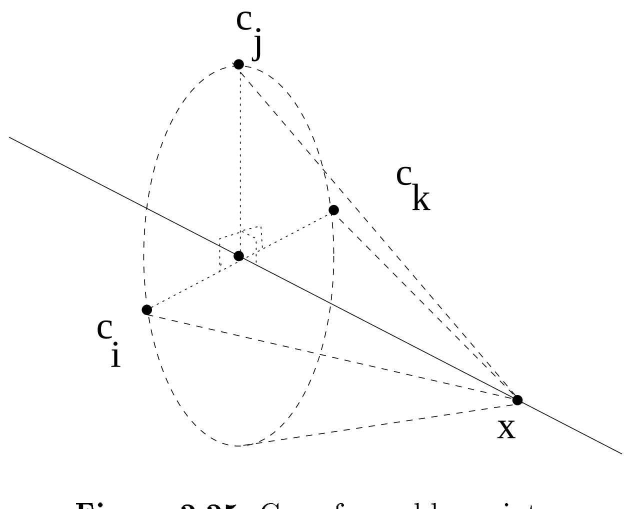

Figure 2.25 Cone formed by points.

Now, let *gradient plane* be the $(k-1)$-dimensional plane that contains the heads of the k gradient vectors based at x. This is the affine hull of the heads of k gradient vectors. Note that $d_{i_j}(x)$ is the length of the line that connects x and c_j where

$c_j \in \{c_1, \ldots, c_k\}$. Since, $\|x - c_{i_1}\| = \|x - c_{i_j}\|$ for all j, $x - c_{i_j} = \|x - c_{i_1}\|\nabla d_{i_j}(x)$. Therefore, the cone formed by the gradient vectors is "similar" to the one formed by the k closest points and thus the *gradient plane* is parallel to the baseplane. Therefore, the tangent space is perpendicular to the gradient plane. ■

Lemma 2.20 *Let $c_1, \ldots, c_k$ be the closest points in the k nearest obstacles to $x \in \mathcal{F}_{i_1 \ldots i_k}$. At a point x in the k-equidistant face, the tangent space $T_x\mathcal{F}_{i_1 \ldots i_k}$ is the same as the tangent space $T_x\mathcal{R}_{i_1 \ldots i_k}$, where $\mathcal{R}_{i_1 \ldots i_k}$ is the regular k-equidistant face defined by $c_1, \ldots, c_k$.*

Proof: A k-equidistant face can be defined by $G^{-1}(0)$ where

$$G(x) = \begin{bmatrix} (d_1 - d_2)(x) \\ \vdots \\ (d_1 - d_k)(x) \end{bmatrix}.$$

Let $\{c_i\}$ denote the closest points to x in the k closest obstacles. The *regular* k-equidistant face for the set of points $\{c_i\}$ is defined by $VG^{-1}(0)$ where

$$VG(x) = \begin{bmatrix} |x - c_1| - |x - c_2| \\ |x - c_1| - |x - c_3| \\ \vdots \\ |x - c_1| - |x - c_k| \end{bmatrix}.$$

Since the set of closest points, $\{c_i\}$, is the same for the k-equidistant face and the regular k-equidistant face at x, $d_h(x) = |x - c_h|$ for all $h = 1, \ldots, k$. That is, $\nabla(d_{h_1} - d_{h_2})(x) = |x - c_{h_1}| - |x - c_{h_2}|$ for all $h_1, h_2 = 1, \ldots, k$. Therefore, $G(x) = VG(x)$ at x.

The tangent space at a point $x \in G^{-1}(0)$ is the null space of $\nabla G(x)$, where

$$\nabla G(x) = \begin{bmatrix} \nabla(d_1 - d_2)(x) \\ \vdots \\ \nabla(d_1 - d_k)(x) \end{bmatrix}.$$

and the tangent space at a point $x \in VG^{-1}(0)$ is the null space of $\nabla VG(x)$, where

$$\nabla VG(x) = \begin{bmatrix} \frac{x-c_1}{|x-c_1|} - \frac{x-c_2}{|x-c_1|} \\ \frac{x-c_1}{|x-c_1|} - \frac{x-c_3}{|x-c_1|} \\ \vdots \\ \frac{x-c_1}{|x-c_1|} - \frac{x-c_k}{|x-c_1|} \end{bmatrix}.$$

Again, since the set of closest points, $\{c_i\}$, is the same for the k-equidistant face and the regular k-equidistant face at x, $\nabla d_h(x) = \frac{x-c_h}{|x-c_h|}$ for all $h = 1, \ldots, k$. That is, $d_{h_1}(x) - d_{h_2}(x) = \frac{x-c_{h_1}}{|x-c_{h_1}|} - \frac{x-c_{h_2}}{|x-c_{h_2}|}$ for all $h_1, h_2 = 1, \ldots, k$. Therefore, $\nabla G(x) = \nabla VG(x)$ at x. Hence, the k-equidistant face and the regular k-equidistant face have the same tangent space at x. ■

Proposition 2.21 *The restriction of the multi-object distance function D to a k-equidistant face is smooth. That is, the generalized gradient of $D(x)$ projected onto $T_x\mathcal{F}_{i_1\ldots i_k}$ is equal to $\pi_{T_x\mathcal{F}_{i_1\ldots i_k}}\nabla d_i$ for all $i \in \{i_1 \ldots i_k\}$, where π is the orthogonal projection operator.*

Let E be a plane in $T_x\mathbb{R}^m$. Let v_e be the unique minimum length vector in E (i.e., v_e is based at the origin of $T_x\mathbb{R}^m$ and its head is in E). Define P_E to be the subspace of $T_x\mathbb{R}^m$ parallel to E, i.e. $P_E = E - v_e$. Let $P_E^\perp$ be the orthogonal compliment of P_E. Therefore, $T_x\mathbb{R}^m = P_E \bigoplus P_E^\perp$ and thus for all vectors $u \in T_x\mathbb{R}^m$, u can be writen as the sum $u_1 + u_2$ where $u_1 \in P_E$ and $u_2 \in P_E^\perp$. The orthogonal projection $\pi_{P_E}(u)$ is u_1. We can now define the orthogonal projection operator π_E to be π_{P_E}.

Proof: Note that $\partial D(x)$ is the affine hull of the heads of the k gradient vectors $\nabla d_{i_1}, \ldots, \nabla d_{i_k}$. So, $\partial D(x)$ can be viewed as a plane in $T_x\mathbb{R}^m$ and by Lemmas 2.19 and 2.20, the plane $\partial D(x)$ is orthogonal to $T_x\mathcal{SS}_{i_1\ldots i_k}$. Transverality considerations imply that $\partial D(x)$ and $T_x\mathcal{SS}_{i_1\ldots i_k}$ intersect at a point, and thus the generalized gradient of D constrained to $T_x\mathcal{SS}_{i_1\ldots i_k}$ is always a point, not a vector. This point, which we denote by $v \in T_x\mathcal{SS}_{i_1\ldots i_k} \bigcap \partial D(x)$ is the closest point in $\partial D(x)$ to $0 \in T_x\mathbb{R}^m$.

Define P to be a subspace of $T_x\mathbb{R}^m$ given by $P = \partial D(x) - v$ (again, $\partial D(x)$ is viewed as a plane). The orthogonal projection of $u \in \partial D(x)$ is given by

$$\pi_{T_x \mathbb{SS}_{i_1 \ldots i_k}} : \partial D(x) \to T_x \mathbb{SS}_{i_1 \ldots i_k} \tag{2.22}$$

Since D and $\pi_{T_x \mathbb{SS}_{i_1 \ldots i_k}}$ are continuous, the restriction of the generalized gradient of D on $\mathbb{SS}_{i_1 \ldots i_k}$ is continuous. Therefore, the restriction of the multi-object distance function D to a k-equidistant face is smooth. ■

2.3.5 Accessibility

Accessibility is the property that a path can be constructed from any point in the free space to a point on the GVG. In this section, we give an argument that a path exists from any point in the freespace to a GVG edge.

Proposition 2.22 *Given the Boundedness Assumption and the Equidistant Surface Transversality Assumption, the GVG has the property of accessibility.*

Proof: We demonstrate that a robot can access the GVG by following a path that is constructed using gradient ascent on the multi-object distance function $D(x)$, which is the distance to the nearest object from x. In Chapter B, it is shown that D is not smooth, and thus does not have a conventional gradient. However, the multi-object distance function does exhibit a *generalized gradient* (Section B.3) which is denoted

$$\partial D(x) = \mathrm{Co}\{\nabla d_i(x) : \ \forall i \in I(x)\}, \tag{2.23}$$

where $I(x)$ is the set of indices where $d_i(x) = D(x)$, and where Co denotes convex hull. That is, it indexes the obstacles that are equidistant to x.

Furthermore, it is shown in Section B.4 that if $0 \in \mathrm{int}(\partial D(x))$, where 0 is the origin of the tangent space at x, then x is a local maxima of D. Using this result and the following two lemmas whose proofs appear in Section A.3, we can conclude that if x is a local maxima of D, then the point x is equidistant to $m+1$ obstacles.

Lemma A.21 *Given a set of n arbitrary vectors in $\mathbb{R}^m$, then $0 \in \mathrm{int}(\mathrm{Co}\{v_i \in \mathbb{R}^m : i = 1, \ldots, n\})$ if and only if $\{v_i \in \mathbb{R}^m : i = 1, \ldots, n\}$ positively span $\mathbb{R}^m$.*

Lemma A.22 [Goldman and Tucker] *It requires a minimum of $(m+1)$ vectors to positively span $\mathbb{R}^m$.*

The results of Scheimber and Olivera [41] state that the second derivative of D, termed *a generalized hesian*, is always positive or contains only positive values. In a sense, the graph of the function D is always "concave up." Therefore, the generalized gradient of D only vanishes at a local minima. Assume the robot does not start at a local minima; this assumption is reasonable because since we are performing a gradient ascent operation, the local minima are unstable extrema points. That is, if the robot were slightly perturbed from a local minima, it will escape. Since the local minima occur on a set of measure zero, they can be practically ignored. Therefore, gradient ascent of the multi-object distance function will bring the robot to a local maxima of D, which is a point equidistant to $m+1$ obstacles and thereby is a point on the GVG. (Note that when ∂D is a set, the vector with the smallest norm in ∂D is chosen as the gradient [41].) ■

2.3.6 Departibility

Departibility is the property of a roadmap that any point in the free space is accessible from some point in the roadmap [38]. In the case where full knowledge of the world's geometry is available, departability is simply accessibility, but in reverse. The "on-line" case is considered later on in Section 2.3.6.

2.3.7 Connectivity of the GVG

For $m = 2$ (i.e., planar environments), the GVD and the GVG are the same, and thus the GVG is connected because the GVD is connected (Section C.4). Although all GVG's in two-dimensional workspaces and the particular GVG in Figure 2.13 are connected, the GVG is not guaranteed to be connected in all work spaces whose dimension is greater than two. Figure 2.26 contains an example of a disconnected GVG with two connected components: (1) an outer GVG network similar to the one

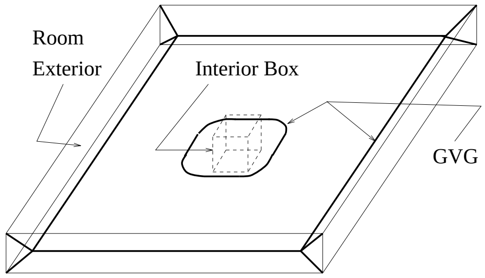

Figure 2.26 An example of a disconnected GVG.

described in Example 2.7 and (2) an inner GVG network which forms a halo-like structure around the inner box.

The generalized Voronoi regions and equidistant faces may be viewed as an exact cellular decomposition (Section 1.2.1) of $\mathcal{W}$ into k-dimensional sets, called *cells*, where $k = 0, \ldots, m$. If each k-dimensional cell is homeomorphic to a k-dimensional disk, then the one-dimensional cells of such a decomposition form a deformation retract or retract-like structure of $\mathcal{W}$ [42]. One of the appealing properties of a retract-like structure is its connectivity.

In the example shown in Figure 2.26, the three-dimensional generalized Voronoi regions are homeomorphic to a three-dimensional disk, but the two-dimensional faces are not homeomorphic to a two-dimensional disk. Furthermore, the "cycle"-like GVG edge surrounding the box is also not homeomorphic to a one-dimensional face.

In some cases, the local connectivity of the GVG may be sufficient or useful for planning. Nevertheless, the GVG by itself is not useful for developing complete motion planning schemes due to this lack of connectivity. In the ensuing sections, we define additional structures, termed the higher order generalized Voronoi graphs, demonstrate some of their properties, and use these properties to show that the union of the GVG and higher order GVG's is connected. We show connectivity in two major steps. First, we make a weak assumption, called the Extended Boundedness Assumption, and show that union of the GVG and higher order GVG's are connected. In current work, we relax this assumption and prescribe some linking strategies which guarantee that the resulting structure is connected.

Chapter 3

The Hierarchichal Generalized Voronoi Graph

The complete roadmap will consist of the one-dimensional GVG augmented by additional one-dimensional structures, termed *higher order generalized Voronoi graphs*. The higher order generalized Voronoi graphs are used to link disconnected GVG components. Essentially, higher order generalized Voronoi graphs are like GVG's that are constrained to equidistant faces. For example, *a second order generalized Voronoi graph*, denoted GVG^2, is analogous to a GVG that is restricted to a two-equidistant face. It is defined in a similar way as the GVG; for example, there are *second order equidistant faces* whose intersections form second order generalized Voronoi edges. An ith order generalized Voronoi graph, denoted GVG^i, is analogous to a GVG on an $(i-1)$st order two-equidistant face. The *hierarchical generalized Voronoi graph* (HGVG) is the GVG and all higher order generalized Voronoi graphs. The underlying philosophy of the HGVG is to exploit the connectivity property of the GVD.

In Section 3.1, we define the hierarchical generalized Voronoi graph. Then, in Section 3.2, we show how cycles give rise to disconnected GVG's. However, in Section 3.3, we state an assumption under which cycles do not occur, and thus the HGVG is connected, as described in Section 3.4.

3.1 The Second Order GVG

Recall from Section 2.2.1 that the GVD is the union of all two-equidistant faces in $\mathcal{W}$. For the moment consider the case of $\mathcal{W} \subset \mathbb{R}^3$, where

1. the only higher order generalized Voronoi graph is the GVG^2,
2. a two-equidistant face is two-dimensional, and
3. a GVG edge is a three-equidistant face that is formed by the intersection of *three* two-equidistant faces.

By definition, a GVG edge lies on the boundary of a two-equidistant face, and thus adjacent two-equidistant faces share a common GVG edge. If the GVG edges associated with each two-equidistant face are connected (i.e., the boundary of each two-equidistant face is connected), then the entire GVG is connected because the GVD is connected. A disconnected GVG may arise when a two-equidistant face contains disconnected boundary components, like the two-equidistant face defined by the floor and the ceiling depicted in Figure 2.26. The GVG^2 is used to connect the boundaries of two-equidistant faces with disconnected boundary components, and thereby connect all disconnected GVG components.

In the ensuing sections, we show that given a condition, introduced in Section 3.3, on each two-equidistant face, $\mathcal{F}_{ij}$, the second order GVG, restricted to $\mathcal{F}_{ij}$, connects the boundaries of $\mathcal{F}_{ij}$, thus linking the GVG. We denote the second order GVG restricted to $\mathcal{F}_{ij}$ as $\mathrm{GVG}^2\Big|_{\mathcal{F}_{ij}}$.

3.1.1 Second Order Generalized Voronoi Region

A $\mathrm{GVG}^2\Big|_{\mathcal{F}_{ij}}$ is a network of one-dimensional curves that divides a two-equidistant face, $\mathcal{F}_{ij}$, into sub-regions where there exists a common *second closest* object (C_i and C_j are the closest objects). These regions are called *second order generalized Voronoi regions* and are defined as

$$\mathcal{F}_k\Big|_{\mathcal{F}_{ij}} = \mathrm{cl}\{x \in \mathcal{F}_{ij} : \forall h \neq i,j,k, \quad 0 < d_i(x) = d_j(x) < d_k(x) < d_h(x) \text{ and } \nabla d_i(x) \neq \nabla d_j(x)\}. \tag{3.1}$$

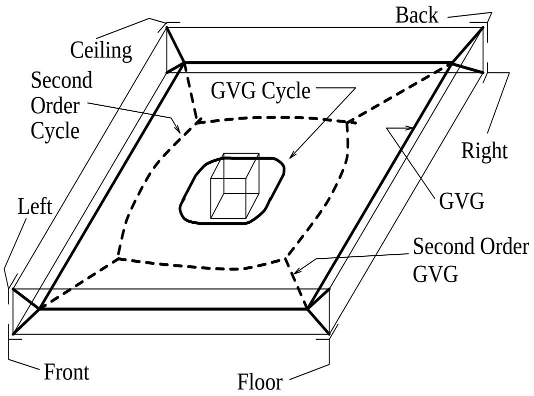

Figure 3.1 Box in a room.

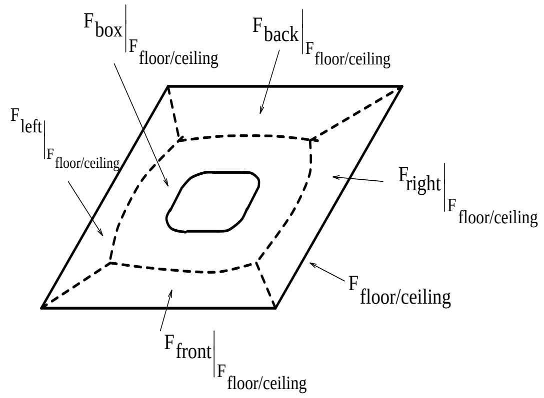

Figure 3.2 Two-equidistant face, defined by the floor and ceiling, is divided up into sub-regions which have a common second closest object.

Example 3.1 Let $\mathrm{GVG}^2\Big|_{\mathcal{F}_{floor/ceiling}}$ be the second order GVG for the two-equidistant face, $\mathcal{F}_{floor/ceiling}$, defined by the floor and ceiling of the rectangular enclosure in Figure 3.1. The solid lines in Figure 3.1 represent the GVG and the dotted lines represent $\mathrm{GVG}^2\Big|_{\mathcal{F}_{floor/ceiling}}$. The $\mathrm{GVG}^2\Big|_{\mathcal{F}_{floor/ceiling}}$ divides $\mathcal{F}_{floor/ceiling}$ into five regions whose closest obstacles are the floor and ceiling; however, each region has a unique second closest obstacle: the front face, the right face, the back face, the left face, and the interior box (Figure 3.2). These regions are the second order generalized Voronoi regions. ♦

The second order generalized Voronoi regions are defined in terms of a distance function. We use the following naming convention: the name of the distance function precedes the name of the GVG^2 region. For example, the *visible second order generalized Voronoi region* is defined as

$$\mathcal{F}_k^V\Big|_{\mathcal{F}_{ij}} = \mathrm{cl}\{x \in \mathcal{F}_{ij} : \ \forall h \neq i, j, k, \quad 0 < d_i^V(x) < d_k^V(x) < d_h^V(x) \text{ and } \nabla d_i^V(x) \neq \nabla d_j^V(x)\}. \tag{3.2}$$

In future discussion, the superscripts are omitted when it is obvious which distance function is being used. Throughout this work, we will use the visible distance function, so the V-superscript will be omitted.

It will be shown that on each two-equidistant face, $\mathcal{F}_{ij}$, the $\mathrm{GVG}^2\Big|_{\mathcal{F}_{ij}}$ connects the boundaries of $\mathcal{F}_{ij}$ if and only if the boundaries of the individual second order generalized Voronoi regions are connected (or can be readily connected with a link). Therefore, our goal is to demonstrate that the boundaries of the individual second order generalized Voronoi regions are connected. The rest of this section is now devoted to careful consideration of the boundary components of the *visible* second order generalized Voronoi regions (and thus we will drop the V superscript). These boundary structures are defined in terms of line of sight information and can be incrementally constructed using range sensor data.

Inspection of Equation 3.2 yields the boundary components of a second order generalized Voronoi region. Starting from the left, consider the first inequality, $0 < d_i(x) = d_j(x)$. The boundary associated with this inequality is the set of points where $0 = d_i(x) = d_j(x)$; this corresponds to a *boundary edge.* Consider the next inequality, $d_i(x) = d_j(x) < d_j(x)$. The boundary associated with this inequality is the set of points, $d_i(x) = d_j(x) = d_k(x)$; this corresponds to a *GVG edge.* The next inequality, $d_k(x) < d_h(x)$, is associated with a common boundary of two adjacent second order generalized Voronoi regions. When the distance to the second closest obstacle continuously changes as a robot crosses from one region to another (i.e., $d_i(x) = d_j(x) < d_k(x) = d_h(x)$), the corresponding structure is a *GVG^2 equidistant edge.* When the distance to the second closest obstacle does *not* continuously change, the corresponding structure is an occluding boundary edge. The final boundary structure is when the two gradients first become collinear ($\nabla d_i(x) = \nabla d_j(x)$); this structure is a *floating boundary edge.*

The following proposition guarantees that in $\mathbb{R}^3$, the only structures on the boundary of a second order generalized Voronoi region $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ may be

- a GVG edge $\mathcal{F}_{ijk}$,

- GVG^2 equidistant edges $\bigcup_l \mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$,
- a boundary edge C_{ij},
- a floating boundary edge F_{ij}, and
- an occluding edge.

Proposition 3.2 *The boundary of a second order generalized Voronoi region may contain the following structures: at most one two-equidistant face, a second order two-equidistant faces, a two-boundary face, a floating two-boundary face, and a two-occluding face.*

Proof: This proof uses the following lemma and proposition:

Lemma A.11 *For two subsets A and B of $\mathbb{R}^m$, the closure of the intersection of A and B is a subset of the intersection of the closures of A and B. In other words,*

$$\mathrm{cl}(A \bigcap B) \subset \mathrm{cl}(A) \bigcap \mathrm{cl}(B).$$

Proposition A.10 *For two subsets A and B of $\mathbb{R}^m$, the boundary of the intersection of the two sets obeys the following inclusion:*

$$\partial(A \bigcap B) \subset \left(\partial A \bigcap \mathrm{cl}(B)\right) \bigcup \left(\partial B \bigcap \mathrm{cl}(A)\right).$$

The definition of a second order generalized Voronoi region is $\mathrm{cl}\{x \in \mathcal{W} : 0 < d_i(x) = d_j(x) < d_k(x) < d_h(x) \forall h$ and $\nabla d_i(x) \neq \nabla d_j(x)\}$. The second order generalized Voronoi region $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ can also be defined as $A \bigcap B \bigcap C \bigcap D$ where

$$\begin{aligned} A &= \mathrm{cl}\{x \in \mathcal{W} : d_i(x) = d_j(x) < d_k(x)\}, \\ B &= \mathrm{cl}\{x \in \mathcal{W} : d_k(x) < d_h(x)\, \forall h\}, \\ C &= \mathrm{cl}\{x \in \mathcal{W} : D(x) > 0\}, \\ D &= \mathrm{cl}\{x \in \mathcal{W} : \nabla d_i(x) \neq \nabla d_j(x)\}. \end{aligned} \tag{3.3}$$

By Proposition A.10,

$$\begin{aligned}\partial(A \bigcap B \bigcap C \bigcap D) \subset \quad & \partial A \bigcap \mathrm{cl}(B) \bigcap \mathrm{cl}(C) \bigcap \mathrm{cl}(D) \\ & \bigcup \partial B \bigcap \mathrm{cl}(A) \bigcap \mathrm{cl}(C) \bigcap \mathrm{cl}(D) \\ & \bigcup \partial C \bigcap \mathrm{cl}(A) \bigcap \mathrm{cl}(B) \bigcap \mathrm{cl}(D) \\ & \bigcup \partial D \bigcap \mathrm{cl}(A) \bigcap \mathrm{cl}(B) \bigcap \mathrm{cl}(C).\end{aligned} \tag{3.4}$$

It can be seen that three of the four components of $\partial(A \bigcap B \bigcap C \bigcap D)$ can be identified:

$$\begin{aligned}\partial A \bigcap \mathrm{cl}(B) \bigcap \mathrm{cl}(C) \bigcap \mathrm{cl}(D) &= \mathcal{F}_{ijk}, \\ \partial C \bigcap \mathrm{cl}(A) \bigcap \mathrm{cl}(B) \bigcap \mathrm{cl}(D) &= C_{ij}, \\ \partial D \bigcap \mathrm{cl}(A) \bigcap \mathrm{cl}(B) \bigcap \mathrm{cl}(C) &= FC_{ij}.\end{aligned} \tag{3.5}$$

The fourth component $\partial B \bigcap \mathrm{cl}(A) \bigcap \mathrm{cl}(C) \bigcap \mathrm{cl}(D)$ corresponds to structures on the boundary of adjacent second order generalized Voronoi regions. Since we are considering second order generalized Voronoi regions defined in terms of the visible distance functions, $\partial B \bigcap \mathrm{cl}(A) \bigcap \mathrm{cl}(C) \bigcap \mathrm{cl}(D)$ comprises both second order two-equidistant faces and occluding two-faces because

$$\begin{aligned}\partial B \quad &= \{x \in \mathcal{W} : d_k(x) = d_h(x) \text{ for some } h\} \\ & \quad \bigcup \{x \in \mathcal{W} : d_k(x) \neq d_h(x) \text{ and } \mathrm{cl}(\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}) \bigcap \mathrm{cl}(\mathcal{F}_h\Big|_{\mathcal{F}_{ij}}) \neq \emptyset \text{ for some } h\}, \\ \mathrm{cl}(A) \quad &= \{x \in \mathcal{W} : d_i(x) = d_j(x) \leq d_k(x)\}, \\ \mathrm{cl}(C) \quad &= \{x \in \mathcal{W} : D(x) \geq 0\}, \\ \mathrm{cl}(D) \quad &= \mathrm{cl}\{x \in \mathcal{W} : \nabla d_i(x) \neq \nabla d_j(x)\}.\end{aligned} \tag{3.6}$$

In other words, $\partial B \bigcap \mathrm{cl}(A) \bigcap \mathrm{cl}(C) \bigcap \mathrm{cl}(D)$ represents the set of points on the shared boundary of adjacent second order generalized regions. The GVG^2 equidistant edges correspond to those points where $d_k(x) = d_l(x)$ and the occluding edges correspond to those points where $d_k(x) \neq d_l(x)$. ■

The next two subsections describe in more detail the GVG^2 equidistant edges and the occluding edges.

3.1.2 Second Order Equidistant Face

The construction of the GVG^2 parallels that of the GVG. The basic building block of the GVG^2 is called the *second order two-equidistant surface* and is defined as $\mathcal{S}_{kl}\Big|_{\mathcal{F}_{ij}} = \{x \in \mathcal{F}_{ij} : (d_l - d_k)(x) = 0\}$. Of particular interest is the subset of $\mathcal{S}_{kl}\Big|_{\mathcal{F}_{ij}}$ termed the *two-equidistant surjective surface*, which is defined (for convex obstacles) as $\mathcal{SS}_{kl}\Big|_{\mathcal{F}_{ij}} = \{x \in \mathcal{S}_{kl}\Big|_{\mathcal{F}_{ij}} : \nabla d_l(x) \neq \nabla d_k(x)\}$. We define the *second order two-equidistant face* to be

$$\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}} = \{x \in \mathrm{cl}(\mathcal{SS}_{kl}\Big|_{\mathcal{F}_{ij}}) \quad \text{such that} \quad \forall h,\, d_h(x) \geq d_k(x) = d_l(x) \geq d_i(x) = d_j(x)\}. \tag{3.7}$$

The second order two-equidistant face, $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$, is the set of points on the face, $\mathcal{F}_{ij}$, that are equidistant to two obstacles C_k and C_l such that C_k and C_l are the *second* closest equidistant objects and C_i and C_j are the closest equidistant obstacles. See Figure 3.3 for an example of second order two-equidistant faces, which are drawn as dotted lines.

Analogous to the GVG, we continue our construction with lower dimensional subsets of $\mathcal{F}_{ij}$. The *second order three-equidistant face*,

$$\mathcal{F}_{klp}\Big|_{\mathcal{F}_{ij}} = \mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}} \bigcap \mathcal{F}_{lp}\Big|_{\mathcal{F}_{ij}} \bigcap \mathcal{F}_{kp}\Big|_{\mathcal{F}_{ij}},$$

is the set of points where C_k, C_l, and C_p are *second* closest equidistant objects and C_i and C_j are the closest equidistant objects.

The second order k-equidistant face is the intersection of the appropriate $k-3$ second order two-equidistant faces. In m-dimensions, the *second order generalized Voronoi edge* (GVG^2 equidistant edge) is a second order $(m-1)$-equidistant face. Note, it is defined by $m-3$ intersections of second order two-equidistant faces, whereas the (first order) generalized Voronoi edge is defined by $m-2$ intersections of (first order) two-equidistant faces. It can be easily shown by the pre-image

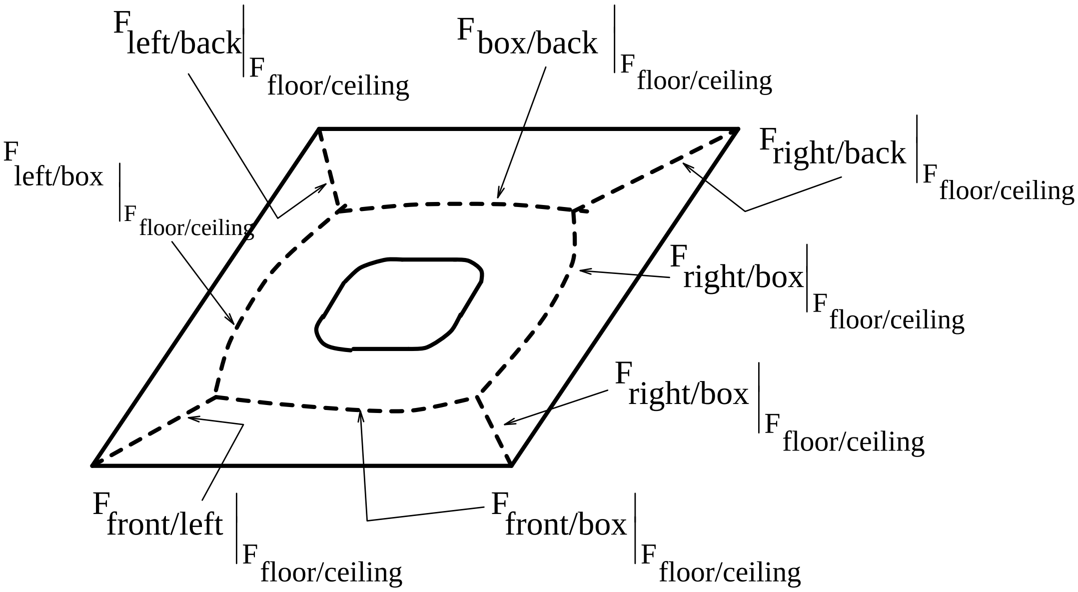

Figure 3.3 Second order two-equidistant faces are drawn in dotted lines. These are the set of points equidistant to their first two and second two closest objects. Note that this figure is identical to Figure 3.2, except that the two-equidistant faces are labeled instead of the second order generalized Voronoi regions.

theorem that the GVG^2 equidistant edges are one-dimensional. Finally, a *second order generalized Voronoi vertex* is a second order m-equidistant face, and it is zero-dimensional. Second order generalized Voronoi vertices are also be referred to as *second order meet points.*

In $\mathbb{R}^3$, the second order two-equidistant faces are the GVG^2 equidistant edges (second order generalized Voronoi edges), and the second order three-equidistant faces are the second order generalized Voronoi vertices (or second order meet points). The dotted lines in Figures 3.2 and 3.3 are second order generalized Voronoi edges. They intersect at second order meet points.

In Figure 3.3, the second order GVG only has GVG^2 equidistant edges, but the second order GVG may contain other structures. These structures are boundary faces, defined in Section 2.2.3; floating boundary faces, defined in Section 2.2.3; and *occluding faces*, defined below.

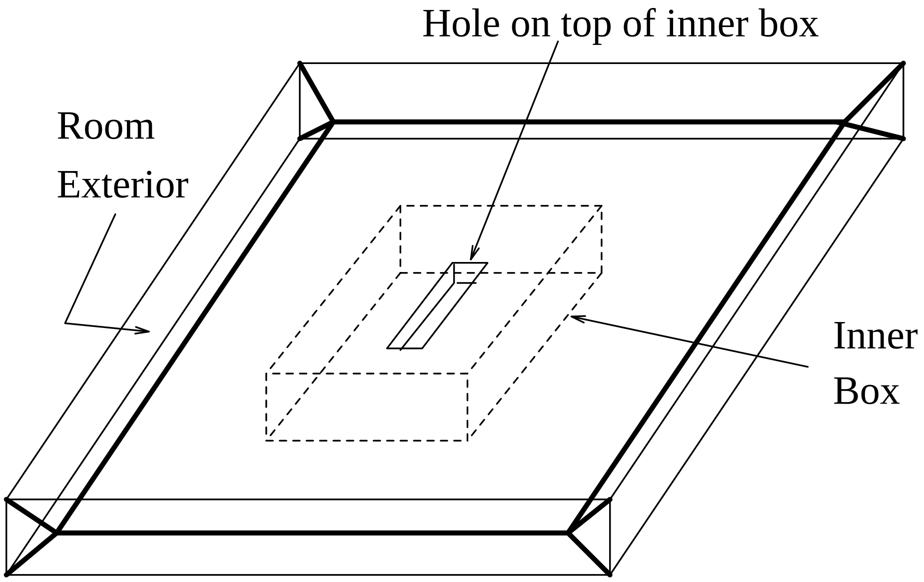

Figure 3.4 Room with a box in the middle. The box, outlined with dotted lines, has an opening on top of it, delineated with solid lines.

3.1.3 Occluding Face

Example 3.3 (Hole on top of a Box) Figure 3.4 depicts a flat room with a box in the middle of the room. The box in the middle of the room contains an opening which can either be a through-hole, a dimple, or an entrance to another internal environment.

The GVG structure associated with the box and the hole (see Figure 3.5) contains two connected components: one associated with the hole and ceiling, and one associated with the box, the floor, and the ceiling. Unfortunately, the two connected components are not within line of sight of each other. Hence, depending upon the robot's initial conditions, it may "miss" one of these connected components while incrementally constructing the HGVG. Therefore, there is a need to define an additional structure to link the disconnected connected components.

♦

We define the occluding face to connect disconnected GVG components that are not within line of sight of each other. In order to define the occluding face, we define

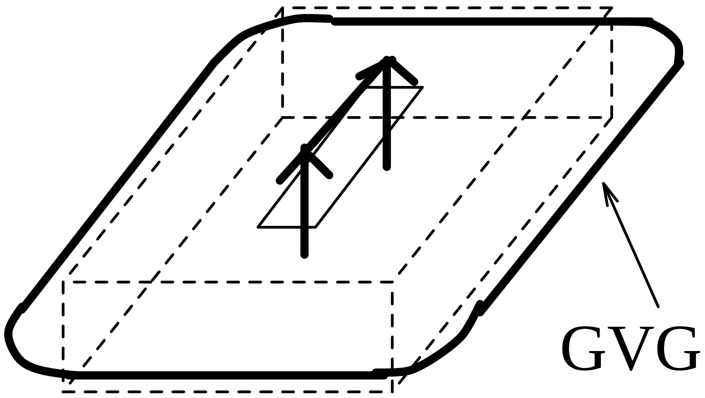

Figure 3.5 The GVG edges in the vicinity of the interior box. This halo-shaped GVG edge is defined by the ceiling, floor and box. The two parallel arrow-like structures connected by a segment is the GVG structure defined by the four sides of the hole and the ceiling.

the *second closest multi-object distance function*, $D^2(x)\Big|_{\mathcal{F}_{ij}} = \min_{k\neq i,j} d_k(x)$. This is the distance to the second closest obstacle to x where x is restricted to be on a two-equidistant face, $\mathcal{F}_{ij}$. Naturally, there exists a visible, x-ray, and slanted second closest multi-object distance functions respectively denoted by $D^{2V}\Big|_{\mathcal{F}_{ij}}$, $D^{2X}\Big|_{\mathcal{F}_{ij}}$ and $D^{2S}\Big|_{\mathcal{F}_{ij}}$. The superscripts will be omitted in situations where it is obvious which distance function is being used. As stated earlier, in this chapter we are using the visible distance function, and thus $D^2(x)\Big|_{\mathcal{F}_{ij}}$ corresponds to the visible multi-object distance function.

Definition 3.4 (Occluding Two-Face) *The occluding two-face, $V_{kl}\Big|_{\mathcal{F}_{ij}}$, is the set of points x on the shared boundary of two adjacent second order generalized Voronoi regions, $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ and $\mathcal{F}_l\Big|_{\mathcal{F}_{ij}}$, where for $s \in \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ and $t \in \mathcal{F}_l\Big|_{\mathcal{F}_{ij}}$, $\lim_{s\to x} d_k(s) \neq \lim_{t\to x} d_l(t)$. That is, it is the set of points where $D^2\Big|_{\mathcal{F}_{ij}}$ is not continuous.*

The existence of an occluding two-face depends upon the choice of distance function. When the second order generalized Voronoi regions are defined in terms of the X-distance function, then there are no occluding two-faces because D^{2X} is always continuous on $\mathcal{F}_{ij}$. Nevertheless, in this work, the second order generalized

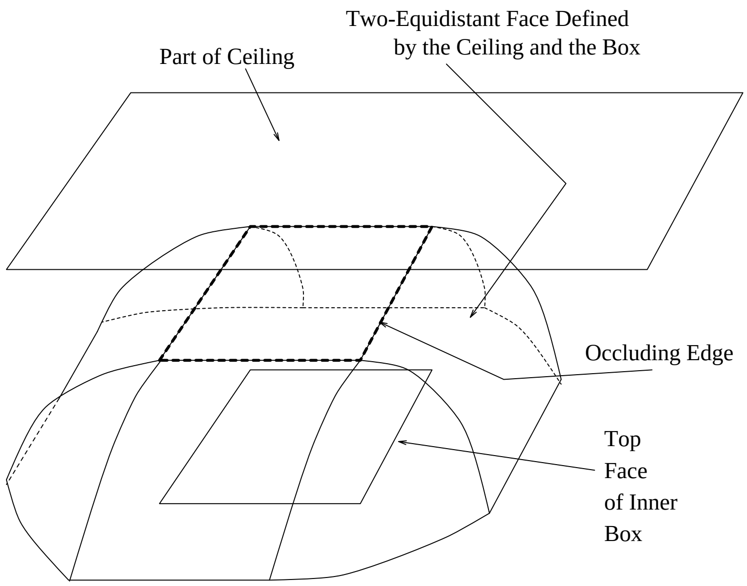

Figure 3.6 Two-equidistant face between the box and the ceiling (from Figure 2.26) is outlined with thin solid lines. All of the enclosure and box from Figure 2.26 is removed with the exception of the top of the box and the ceiling of the enclosure. Dashed lines delineate an occluding edge.

Voronoi regions are defined in terms of visible distance function. Therefore, $D^{2V}\Big|_{\mathcal{F}_{ij}}$ may lose continuity on $\mathcal{F}_{ij}$, and thus give rise to an occluding two-face. In $\mathbb{R}^3$, an occluding two-face is called an *occluding edge.*

Example 3.5 (Occluding Edge) Recall the rectangular enclosure with a box in its interior in Figure 3.1. Consider the two-equidistant face defined by the box and the ceiling of Figure 3.1. This two-equidistant face is shaped like an upside down bowl, as depicted in Figure 3.6. Figure 3.7 contains a side view of Figure 3.6.

Consider a robot in Figure 3.7 that moves from left to right while maintaining double equidistant between the inner box and ceiling (i.e., while it remains on a two-equidistant face). Assume the robot starts at a point where the second closest obstacle is the floor. While moving from left to right on the two-equidistant face, the inner box begins to occlude the floor as the robot begins to pass over the box. (Recall that we are using the visible distance function.) When the floor first becomes occluded, there is a discontinuous jump in the value of the distance to the second

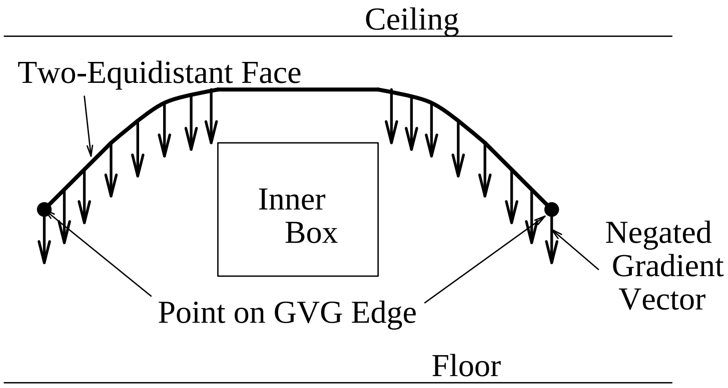

Figure 3.7 The thick solid line represents a side view of the equidistant face defined by the box and the ceiling. The thick arrows that are distributed along the face point towards the floor which is the second closest obstacle. There are no arrows on the portion of the face above the box because the box occludes the floor in that region.

closest obstacle. The point where the floor first becomes occluded is therefore a point in an occluding edge.

The dashed lines in Figure 3.6 represent the occluding edge in the two-equidistant face defined by floor and ceiling. The occluding edge encloses a region where points in its exterior are within line of sight of the floor. (See Figure 3.8.) ⧫

Example 3.6 (Hole on top of Box (continued)) Recall the environment from Example 3.3 which is a rectangular enclosure with a box in its interior. On top of the box, there is an opening which could be a through-hole, an entrance to a sub-environment or a dimple. See Figures 3.4 and 3.5.

Since in this example we are only interested in the GVG edges associated with the box, Figure 3.9 contains only the box, the GVG structures associated with it (thick solid lines) and an occluding edge (thick dotted lines).

The GVG structure associated with the hole is connected to the occluding edge using GVG^2 equidistant edges. Using a linking procedure described later in Section 3.4.2, the outer GVG is linked to the occluding edge. The result is that the GVG is now connected through a link, an occluding edge and an GVG^2 equidistant edge. See Figure 3.10. ⧫

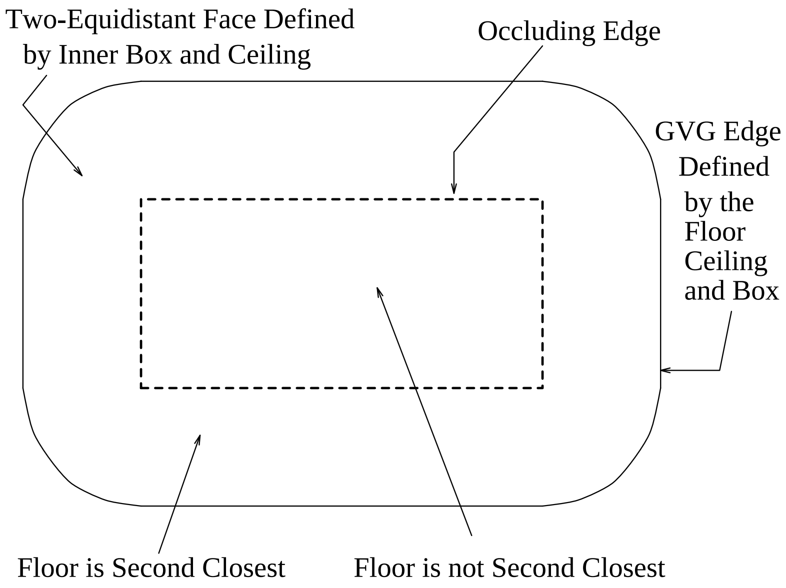

Figure 3.8 Two-equidistant face between the box and the ceiling, as viewed from above, is drawn with an occluding edge.

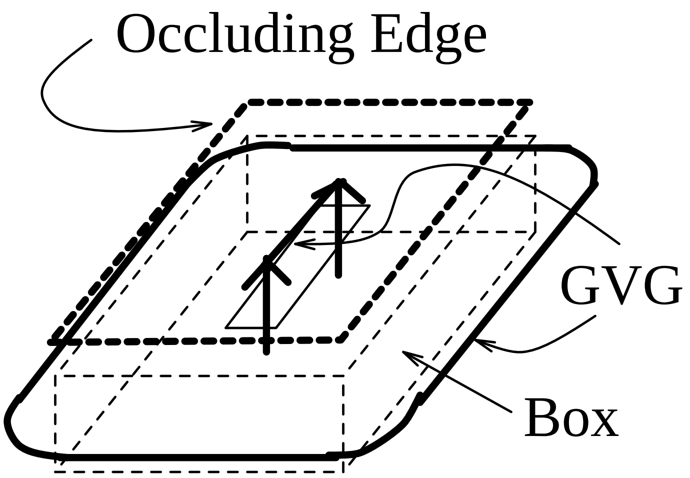

Figure 3.9 The occluding edge, represented by a thick dotted line, is defined by the visible-distance function. The GVG is represented by the thick solid lines and the inner box is drawn in thin dashed lines.

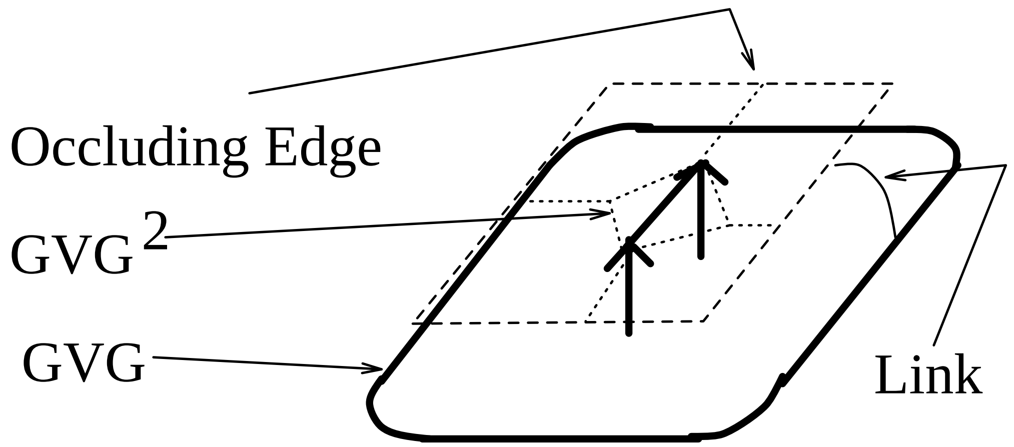

Figure 3.10 The GVG surrounding the box is connected to the GVG associated with the hole through a link from the GVG surrounding the box to the occluding edge. The box is removed for clarity.

3.1.4 Second Order Generalized Voronoi Set

Analogous to the two-Voronoi set which was used to define the GVG, we now define the *second order two-Voronoi set* as the union of second order two-equidistant faces, two-boundary faces, floating two-boundary faces, and two-occluding faces on a two-equidistant face. (Note that there exists two-equidistant faces for which some of these structures do not exist.) When a two-equidistant face, $\mathcal{F}_{ij}$, contains all four structures, we can write the second order two-Voronoi set as

$$\mathfrak{F}^2\Big|_{\mathcal{F}_{ij}} = \bigcup_k \left(\bigcup_l \left(\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}} \bigcup C_{ij} \bigcup FC_{ij} \bigcup V_{kl}\Big|_{\mathcal{F}_{ij}} \right) \right). \tag{3.8}$$

For example, if the $\mathfrak{F}^2\Big|_{\mathcal{F}_{ij}}$ is composed of three second order two-equidistant face and one two-boundary face, the $\text{GVG}^2\Big|_{\mathcal{F}_{ij}}$ is $\bigcup_{p=1}^{p=3} \mathcal{F}_{kl_p} \bigcup C_{ij}$ which is equal to $\mathcal{F}_{kl_1}\Big|_{\mathcal{F}_{ij}} \bigcup \mathcal{F}_{kl_2}\Big|_{\mathcal{F}_{ij}} \bigcup \mathcal{F}_{kl_3}\Big|_{\mathcal{F}_{ij}} \bigcup C_{ij}$.

Continuing in the same vein as the GVG, we can define the *second order three-Voronoi set* as the union of all the second order three-equidistant faces, three-occluding faces, three-boundary faces, and floating three-boundary faces. It is written as

$$\mathcal{F}^3\Big|_{\mathcal{F}_{ij}} = \bigcup_k \left(\bigcup_l \left(\bigcup_p \left(\mathcal{F}_{klp}\Big|_{\mathcal{F}_{ij}} \bigcup C_{ijk} \bigcup FC_{ijk} \bigcup V_{klp}\Big|_{\mathcal{F}_{ij}} \right) \right) \right). \tag{3.9}$$

The *second order k-Voronoi set* is the union of the second order k-equidistant faces, k-boundary faces, floating k-boundary faces, and k-occluding faces. In $\mathbb{R}^m$, a second order GVG is the collection of second order $(m-1)$-Voronoi sets and m-Voronoi sets, i.e., $\mathrm{GVG}^2\Big|_{\mathcal{F}_{ij}} = (\mathcal{F}^{m-1}\Big|_{\mathcal{F}_{ij}}, \mathcal{F}^m\Big|_{\mathcal{F}_{ij}})$. So, when $m = 3$, $\mathrm{GVG}^2\Big|_{\mathcal{F}_{ij}} = (\mathcal{F}^2\Big|_{\mathcal{F}_{ij}}, \mathcal{F}^3\Big|_{\mathcal{F}_{ij}})$.

Definition 3.7 (Hierarchical Generalized Voronoi Graph) The union of the generalized Voronoi graph and higher order Voronoi graphs is termed the *hierarchical generalized Voronoi graph* (HGVG).

In summary, when $m = 3$, the HGVG is the union of the GVG and GVG^2 and contains the following components:

Symbol	Name of Structure	Distance Relationships for x in Structure
$\mathcal{F}_i$	generalized Voronoi region	$d_i(x) \leq d_h(x)\, \forall h$
$\mathcal{F}_{ij}$	generalized Voronoi face	$d_i(x) = d_j(x) \leq d_h(x)\ \forall h$
$\mathcal{F}_{ijk}$	generalized Voronoi edge	$d_i(x) = d_j(x) = d_k(x) \leq d_h(x)\ \forall h$
$\mathcal{F}_{ijkl}$	meet point	$d_i(x) = d_j(x) = d_k(x) = d_l(x) \leq d_h(x)\ \forall h$
$\mathcal{F}_k\Big\vert_{\mathcal{F}_{ij}}$	second order generalized Voronoi region	$d_i(x) = d_j(x) \leq d_k(x) \leq d_h(x)\ \forall h$
$\mathcal{F}_{kl}\Big\vert_{\mathcal{F}_{ij}}$	GVG^2 equidistant edge	$d_i(x) = d_j(x) \leq d_k(x) = d_l(x) \leq d_h(x)\ \forall h$
$\mathcal{F}_{klp}\Big\vert_{\mathcal{F}_{ij}}$	second order meet point	$d_i(x) = d_j(x) \leq d_p(x) = d_k(x) = d_l(x) \leq d_h(x)\ \forall h$
C_{ij}	boundary edge	$d_i(x) = d_j(x) = 0$
FC_{ij}	floating boundary edge	$0 < d_i(x) = d_j(x) \leq d_h(x)$ such that $\nabla d_i(x) = \nabla d_j(x)$
$V_{kl}\Big\vert_{\mathcal{F}_{ij}}$	occluding edge	$d_i(x) = d_j(x) < d_k(x) < d_l(x) < d_h(x)\ \forall h$ or $d_i(x) = d_j(x) < d_l(x) < d_k(x) < d_h(x)\ \forall h$

3.1.5 Connectivity of the GVG and Second Order GVG

The following lemma asserts that the GVG^2 intersects the GVG at meet points. Therefore, when constructing the HGVG, the robot simply constructs a GVG fragment and then from each meet point on that fragment, in constructs a second order GVG which will be connected to other GVG fragments.

Lemma 3.8 *If a second order two-equidstant face $\mathcal{F}_{kl}\Big\vert_{\mathcal{F}_{ij}}$ is not empty, then it is the only second order two-equidstant face which emanates from a four-equidistant face $\mathcal{F}_{ijkl}$ in the two-equidisant face $\mathcal{F}_{ij}$.*

In $\mathbb{R}^3$, this lemma can be restated as follows: if the meet point $\mathcal{F}_{ijkl}$ exists, then there must exist an associated one-dimensional GVG^2 edge $\mathcal{F}_{kl}\Big\vert_{\mathcal{F}_{ij}}$.

Proof: Since $\mathcal{F}_{ijkl} \neq \emptyset$, there must exist at least one point in $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$. Now, it needs to be shown that $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$ is truly one-dimensional.

From the above relations, it can be seen that $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}} \subset \mathcal{SS}_{kl}\Big|_{\mathcal{F}_{ij}} \subset \mathcal{S}_{kl}\Big|_{\mathcal{F}_{ij}}$. Therefore, $\mathcal{F}_{ijkl} \subset \mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}} \Rightarrow \mathcal{F}_{ijlk} \subset \mathcal{SS}_{kl}\Big|_{\mathcal{F}_{ij}}$. For any nbhd$(\mathcal{F}_{ijkl})$, define $Y = (\text{nbhd}(\mathcal{F}_{ijkl}) \bigcap \mathcal{SS}_{kl}\Big|_{\mathcal{F}_{ij}}) \backslash \mathcal{F}_{ijkl}$.

Since Y is an open subset $\mathcal{SS}_{kl}\Big|_{\mathcal{F}_{ij}}$ and the dimension of $\mathcal{SS}_{kl}\Big|_{\mathcal{F}_{ij}}$ is one (by the pre-image theorem), the dimension of Y is also one. For all $y \in Y$, by continuity of the distance function

$$\begin{aligned} d_i(y) < d_h(y), \quad d_j(y) < d_h(y), \\ d_k(y) < d_h(y), \quad d_l(y) < d_h(y). \end{aligned}$$

Furthermore, by definition of Y, $\forall y \in Y$, $d_i(y) = d_j(y)$ and $d_k(y) = d_l(y)$. Therefore, $Y \subset \mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$. Since Y is a one-dimensional subset of $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$, $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$ is one-dimensional. ■

In fact, for each meet point $\mathcal{F}_{ijkl}$ on a two-equidistant face $\mathcal{F}_{ij}$, the corresponding GVG^2 equidistant edge $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$ is unique, as shown by the following:

Lemma 3.9 *Only one second order two-equidistant face emanates from a four-equidistant face.*

Proof: Consider the second order two-equidistant face $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$ which emanates from the four-equidistant face $\mathcal{F}_{ijkl}$ on the two-equidistant face $\mathcal{F}_{ij}$. Assume there exists another second order two-equidistant face $\mathcal{F}_{kp}\Big|_{\mathcal{F}_{ij}}$ that emanates from $\mathcal{F}_{ijkl}$ on $\mathcal{F}_{ij}$. Therefore, $\mathcal{SS}_{kp} \bigcap \mathcal{SS}_{kl} \bigcap \mathcal{F}_{ij} \neq \emptyset$, and thus $\mathcal{SS}_{ijklp} \neq \emptyset$. However, Assumption 2.12 states that $\mathcal{SS}_{ijklp}$ must be empty. Therefore, only one second order two-equidisant may emanate from a four-equidistant face on a two-equidistant face. ■

Using similar arguments as above, it can be shown that (1) a GVG edge may also terminate at a boundary point, (2) a boundary edge terminates at a boundary point

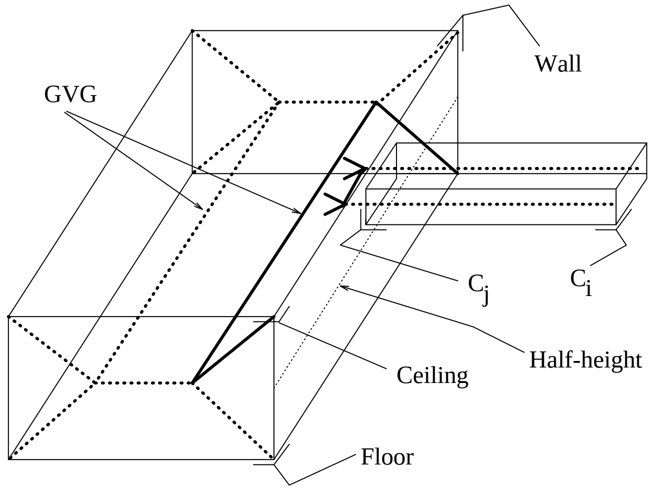

Figure 3.11 A room with a hole in its side wall. The thick dotted lines represent the GVG and the thin dotted line marks the half-height of the room. The thick solid lines are drawn to emphasize the GVG edges associated with the two-equidistant face defined by the right wall and ceiling.

shared by a GVG edge or a boundary point shared by a floating boundary edge, (3) a GVG^2 equidistant edge may also terminate at a second order meet point, a point in the interior of a boundary edge, a point in the interior of a floating bounary edge, or a point in the interior of an occluding edge.

Example 3.10 (A connected HGVG) Figure 3.11 depicts the disconnected GVG for the environment shown in Figure 2.15, from Example 2.10. The geometry of the hole with respect to the room causes the boundary of the two-equidistant face, defined by the wall and the ceiling in Figure 3.11, to be disconnected (Figure 3.12); this results in a disconnected GVG. The second order GVG prescribes a well defined path on the two-equidistant face that connects the disconnected GVG fragments. Therefore, in this example the HGVG is connected. See Figure 3.13.

♦

In many cases, the GVG^2 will link the disconnected components of the GVG. However, as shown in Figure 3.1 (Example 3.1), the GVG^2 may not necessarily serve as a bridge between disconnected GVG components. The HGVG associated with the environment in Figure 3.1 is disconnected because the GVG possesses a *cycle*,

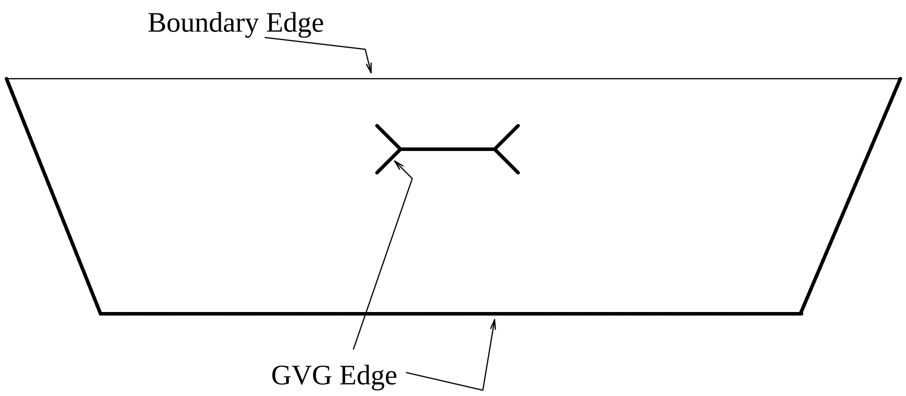

Figure 3.12 GVG edges, drawn as thick solid lines, are on the boundary of the two-equidistant face between the wall and the ceiling of Figure 3.11 in Example 2.10. The GVG structure in the middle of the face is associated with the hole; in actuality, it "pinches up" out of the face.

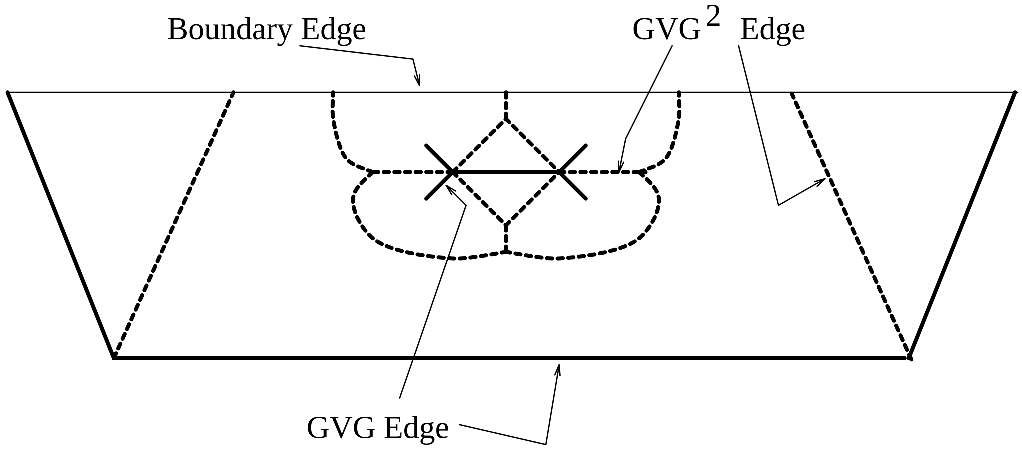

Figure 3.13 The second order GVG edges and boundary edges are drawn in the two-equidistant face between the wall and the ceiling of Figure 3.11 in Example 2.10. The thick solid lines are GVG edges, the dotted lines are GVG^2 edges and the thin solid line is a boundary edge. Here, the GVG^2 links up disconnected GVG edge fragments on the two-equidistant face.

a GVG edge which is always disconnected from other GVG edges.

The next two sections are dedicated to analyzing cycles. Section 3.2 formally defines cycles and details their properties. Section 3.3 contains an assumption which precludes the existence of cycles. Under this assumption, we show in Section 3.4 that the HGVG is connected. Future work includes relaxing this assumption and showing how the HGVG is connected in general (with the inclusion of links).

3.2 Cycles and Periods

The central purpose of this section is to explore one of the reasons why the HGVG is not necessarily connected – the existence of "cycles" in the HGVG such as the one in Figure 3.1.

3.2.1 GVG Cycle

Definition 3.11 (GVG Cycle) A *GVG cycle* is a generalized Voronoi edge which is diffeomorphic to S^1, the unit circle.

Henceforth, the term "cycle" refers to a GVG cycle.

Proposition 3.12 *In a bounded three-dimensional Euclidean space, a GVG edge is a cycle if and only if it is disconnected from the GVG and the* GVG^2.

Proof: This proof is a consequence of the following lemmas whose proof appears in Section 3.2.4.

Lemma 3.13 *When Assumption 2.12 is true, a GVG cycle cannot contain a meet point.*

Lemma 3.14 *A GVG cycle cannot contain any boundary or floating boundary points.*

Lemma 3.15 *In* $\mathbb{R}^3$*, a three-equidistant surface,* $\mathcal{SS}_{ijk}$*, is either diffeomorphic to* S^1 *(i.e., it is a GVG cycle), or it is unbounded.*

Lemma 3.16 *A GVG^2 equidistant edge can only intersect the GVG at a meet point.*

If a GVG edge is a cycle, then it does not contain meet points (Lemma 3.13), boundary points (Lemma 3.14) or floating boundary points (Lemma 3.14), and thus other GVG edges and GVG^2 edges (Lemma 3.16) cannot intersect it. That is, the GVG cycle is disconnected.

Assume there exists a disconnected GVG edge that is not a cycle. By Lemma 3.15, the GVG edge must be unbounded. However, this contradicts our Boundedness Assumption (Assumption 1.1), and thus the GVG edge is a cycle. ∎

A necessary step to show that the HGVG is connected is to prove that the boundaries of each second order generalized Voronoi region are connected (or are readily connected with a link). Whereas, Proposition 3.12 states that the existence of GVG cycle implies that the HGVG is not connected, the next proposition demonstrates how cycles give rise to second order generalized Voronoi regions whose boundaries are not connected.

Proposition 3.17 *In a bounded three-dimensional Euclidean space, a GVG edge is a disconnected component of a boundary of a second order generalized Voronoi region if and only if it is a cycle.*

Proof: This proof is based on the following lemmas, whose results are general in $\mathbb{R}^m$ and whose proofs appear in Section 3.2.4.

Lemma 3.18 *If the three-equidistant face $\mathcal{F}_{ijk}$ is not empty, then the second order generalized Voronoi region $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ must not be empty. Furthermore, if $\mathcal{F}_{ijk} \neq \emptyset$, then $\mathcal{F}_{ijk} \subset \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$.*

Lemma 3.19 *The boundary of a second order generalized Voronoi region contains at most* one *three-equidistant face. That is, $\mathcal{F}_{pqr} \subsetneq \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ for all $\{p,q,r\} \neq \{i,j,k\}$.*

By Lemma 3.18, the GVG edge $\mathcal{F}_{ijk}$ must be a subset of the boundary of a second order generalized Voronoi region, $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. In fact, by Lemma 3.19 it is the only GVG edge that can be in the boundary of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. GVG^2 equidistant edges, boundary edges, floating boundary edges, and occluding edges (Proposition 3.2) are the other structures which *may* exist on the boundary of a second order generalized Voronoi region.

If $\mathcal{F}_{ijk}$ is a cycle, then by Proposition 3.12 none of the above listed structures can intersect it, and thus $\mathcal{F}_{ijk}$ must lie on a disconnected component of the boundary of the second order generalized Voronoi region.

If $\mathcal{F}_{ijk}$ is a disconnected boundary component of a second order generalized Voronoi region, it does not intersect any GVG edge, or any GVG^2 edge. By Proposition 3.12, $\mathcal{F}_{ijk}$ is a cycle. ■

Recall Example 3.1 which consists of a room with a box in its interior. Figure 3.2 shows the two-equidistant face defined by the floor and ceiling. Solid lines represent the GVG and dotted lines represent the GVG^2. The inner box defines a second order generalized Voronoi region, $\mathcal{F}_{box}\Big|_{\mathcal{F}_{floor/ceiling}}$. This region contains a cycle on its boundary and thus has a boundary that is not connected. All of the other second order generalized Voronoi regions do not contain any cycles and thus their boundaries are connected.

3.2.2 Second Order Cycles and Periods

Just as there are cycles in the GVG, there are also cycles in the GVG^2. A *second order cycle* is a second order generalized Voronoi edge (GVG^2 equidistant edge) that is diffeomorphic to S^1, the unit circle. However, we are interested in another structure, termed the *second order period*, defined below.

Definition 3.20 (GVG^2 Period) A *GVG^2 period* is a connected boundary component that does not contain any GVG edges of a second order generalized Voronoi region.

By definition, a GVG^2 period is the union of zero or more GVG^2 equidistant edges, zero or more boundary fragments, zero or more floating boundary fragments, and zero or more occluding fragments. Note that second order periods are homeomorphic to S^1, and GVG^2 cycles are GVG^2 periods.

A GVG^2 period which only has GVG^2 equidistant edges is denoted $\bigcup_l \mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$. A GVG^2 period which has GVG^2 equidistant edges, boundary fragments, floating boundary fragments, and occluding fragments is denoted by

$$\bigcup_l \left(\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}} \bigcup C_{ij} \bigcup FC_{ij} \bigcup V_{kl}\Big|_{\mathcal{F}_{ij}} \right).$$

For example, if a GVG^2 period is composed of three GVG^2 equidistant edges and one boundary edge, the GVG^2 period is $\mathcal{F}_{kl_1}\Big|_{\mathcal{F}_{ij}} \bigcup \mathcal{F}_{kl_2}\Big|_{\mathcal{F}_{ij}} \bigcup \mathcal{F}_{kl_3}\Big|_{\mathcal{F}_{ij}} \bigcup C_{ij}$.

Example 3.21 Figure 3.14 contains a second order period that arises in Example 3.1, which investigates the environment consisting of a room with a box in its interior (Figure 3.1). This environment has many two-equidistant faces, one of which is drawn in Figure 3.2. This is the two-equidistant face defined by the floor and the ceiling. The dotted lines are the second order generalized Voronoi graph. The second order generalized Voronoi region defined by the box ($\mathcal{F}_{box}\Big|_{\mathcal{F}_{floor/ceiling}}$) is shown Figure 3.14. The dotted lines on the outer boundary represent second order generalized Voronoi edges, and the solid line which forms the inner boundary represents a generalized Voronoi edge, which is a cycle. ♦

3.2.3 Inner and Outer Cycles and Periods

Here, we describe the notion of an inner and outer cycle. Recall the corollary to the Jordan curve lemma which states that any closed curve in the plane divides the plane into two regions: one termed the *bounded* section and one termed the *unbounded* section.

Let $\partial_i \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ be a boundary component of the second order generalized Voronoi region, $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. The following two definitions define the *outer* and *inner* boundary components of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ by using the Jordan Curve Lemma, where the boundary

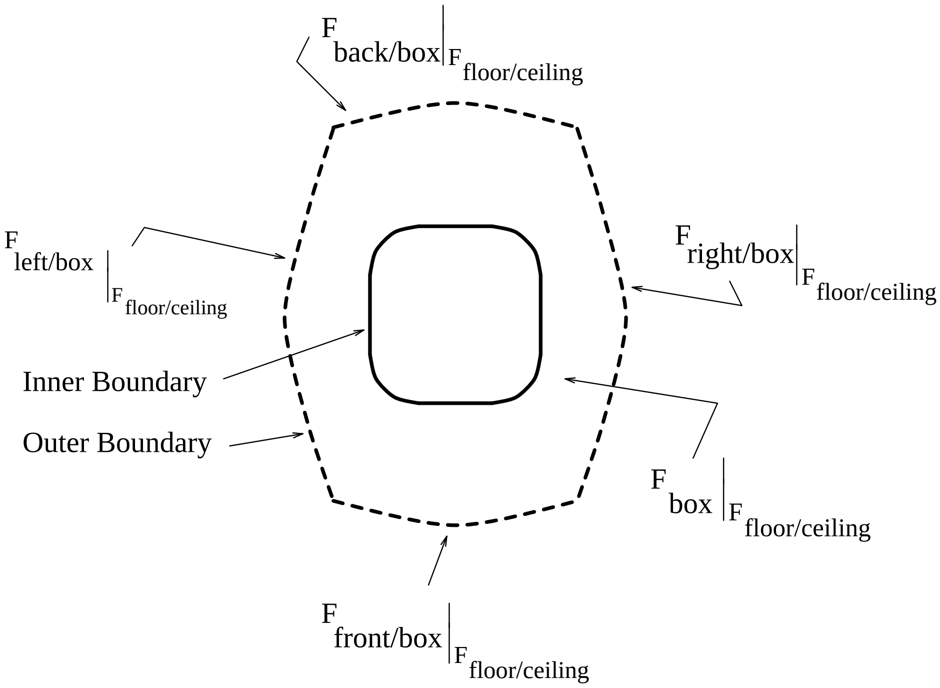

Figure 3.14 The second order period is drawn with dotted lines. It is the union of second order generalized voronoi edges which forms a connected boundary component of a second order generalized Voronoi region.

component, $\partial_i \mathcal{F}_k \Big|_{\mathcal{F}_{ij}}$, serves as the Jordan curve.

Definition 3.22 (Outer Boundary Component) Let $\partial_i \mathcal{F}_k \Big|_{\mathcal{F}_{ij}}$ be a boundary component of the second order generalized Voronoi region, $\mathcal{F}_k \Big|_{\mathcal{F}_{ij}}$. When $\partial_i \mathcal{F}_k \Big|_{\mathcal{F}_{ij}}$ is a Jordon curve on $\mathcal{SS}_{ij}$, it is an *outer boundary component* if $\mathcal{F}_k \Big|_{\mathcal{F}_{ij}}$ lies in the "bounded" region associated with the Jordan curve.

Definition 3.23 (Inner Boundary Component) Let $\partial_i \mathcal{F}_k \Big|_{\mathcal{F}_{ij}}$ be a boundary component of the second order generalized Voronoi region, $\mathcal{F}_k \Big|_{\mathcal{F}_{ij}}$. When $\partial_i \mathcal{F}_k \Big|_{\mathcal{F}_{ij}}$ is a Jordon curve, it is an *inner boundary component* if $\mathcal{F}_k \Big|_{\mathcal{F}_{ij}}$ lies in the "unbounded" region associated with the Jordan curve.

From these two definitions, the notion of an *inner cycle*, *outer cycle*, *inner GVG^2 period*, and *outer GVG^2 period* naturally follow.

Example 3.24 Figure 3.14 contains the second order generalized Voronoi region that is defined by the box on the two-equidistant face, defined by the floor and ceiling from Examples 3.1 and 3.21. The dotted lines in Figure 3.14 represent the GVG^2 period that furnishes the outer boundary. The solid line represents the GVG cycle which is an inner boundary component of $\mathcal{F}_{box} \Big|_{\mathcal{F}_{floor/ceiling}}$. Figures 3.15 and 3.16 illustrate, respectively, how Definitions 3.22 and 3.23 work. In Figure 3.15, when the GVG cycle is a Jordan curve, its associated second order generalized Voronoi region lies in the unbounded region (shaded). Similarly, in Figure 3.16, when the GVG^2 period is a Jordan curve, its associated second order generalized Voronoi region lies in the bounded region (shaded). ♦

From Figures 3.1, 3.2, and 3.14, it appears that there exists a duality between the existence of the GVG cycles and GVG^2 periods. The following proposition establishes this duality: in order for one of them to exist, the other must exist. Hence, the existence of one is a clue to the robot that another cycle or period is

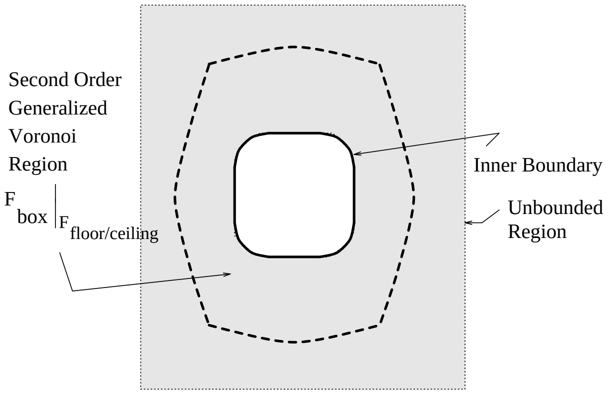

Figure 3.15 Inner boundary.

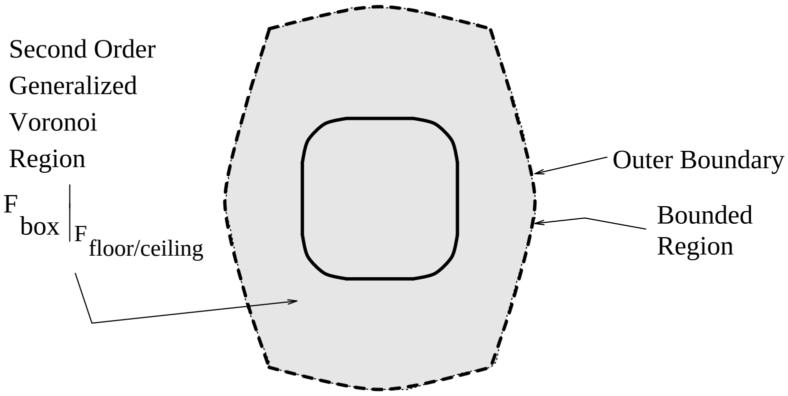

Figure 3.16 Outer boundary.

nearby. This information is needed for a "linking" strategy to connect disconnected HGVG components, such as those in Figure 3.2.

The next three steps are: (1) via the following proposition, show the duality between cycles and second order periods, (2) in the next section, state the assumption under which cycles do not exist (and thus second order periods do not exist), and (3) in the section following the next, show how under this assumption, the HGVG is connected.

Proposition 3.25 *In $\mathbb{R}^3$, if a GVG cycle $\mathcal{F}_{ijk}$ is an inner boundary in a two-equidistant face $\mathcal{F}_{ij}$, then there exists an outer GVG^2 period in the two-equidistant face, $\mathcal{F}_{ij}$.*

Proof: By Lemma 3.18, if $\mathcal{F}_{ijk} \neq \emptyset$, then the second order generalized Voronoi region, $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}} \neq \emptyset$. Furthermore, Lemma 3.18 asserts that $\mathcal{F}_{ijk}$ is in the boundary of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. By Lemma 3.19, $\mathcal{F}_{ijk}$ is the only GVG edge in $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. By the Boundedness Assumption (Assumption 1.1), $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ must be bounded and thus contain an outer boundary component. According to Proposition 3.17, this outer boundary component does not contain $\mathcal{F}_{ijk}$. Such a boundary component is a GVG^2 period because it is free of GVG edges. ■

Although the converse of the above statement is not necessarily true, the following proves to be useful.

Proposition 3.26 *If there exists an* outer *second order period, and there is a generalized Voronoi edge associated with it, then the generalized Voronoi edge is an inner cycle.*

Proof: Recall that a GVG^2 period cannot intersect with a GVG edge. By hypothesis, the GVG^2 period is an outer boundary. Also, by hypothesis, there exists a GVG edge, $\mathcal{F}_{ijk}$, inside the second order period (Figure 3.17).

Assume that the edge $\mathcal{F}_{ijk}$ is not a cycle. If $\mathcal{F}_{ijk} \neq \emptyset$, then $\mathcal{SS}_{ijk} \neq \emptyset$ and by Lemma 3.15 it is unbounded. Therefore, $\mathcal{SS}_{ijk}$ must intersect the outer GVG^2 period

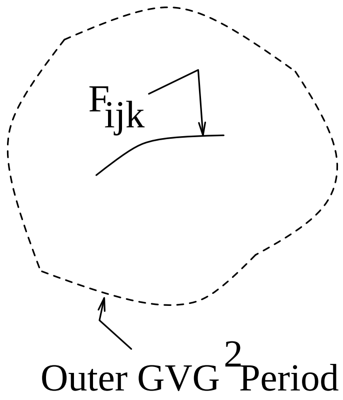

Figure 3.17 *Assume* there exists an outer GVG^2 period with a GVG edge, which is not a cycle, in its interior.

(Figure 3.18). In particular, say $\mathcal{SS}_{ijk}$ intersects $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$. For all $x \in \mathcal{SS}_{ijk} \bigcap \mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$, $d_h(x) \geq d_i(x) = d_j(x) = d_k(x) = d_l(x)$, for all h. This is the definition of a meet point, and thus by Proposition 2.11, a GVG edge intersects $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$. This contradicts our original hypothesis that $\mathcal{F}_{ijk}$ is a GVG^2 period. Therefore, $\mathcal{F}_{ijk}$ is a cycle. ■

The following two implications summarize the previous two propositions.

$$\boxed{\text{Inner GVG cycle}} \quad \Longrightarrow \quad \boxed{\text{Outer GVG}^2\text{ period}}$$

$$\boxed{\text{GVG edge exists and outer second order period}} \quad \Longrightarrow \quad \boxed{\text{GVG cycle}}$$

3.2.4 Proofs of Lemmas Presented in Section 3.2

In the following proofs, recall that nbhd(x) is shorthand for neighborhood of x. Also, recall the following definitions and their distance relationships:

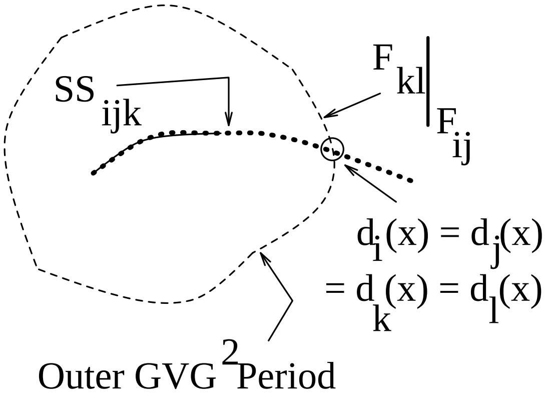

Figure 3.18 If the inner GVG edge is *not* a cycle, then it must intersect the outer GVG^2 period.

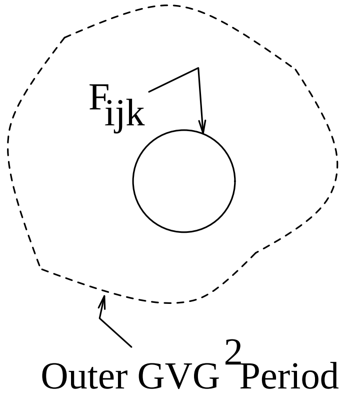

Figure 3.19 The existence of an outer GVG^2 period and a GVG edge associated with it implies there exists inner GVG cycle.

Symbol	Name of Structure	Distance Relationships
$\mathcal{F}_{ij}$	two-equidistant face	$d_i(x) = d_j(x) \leq d_h(x)\ \forall h$
$\mathcal{F}_{ijk}$	GVG edge	$d_i(x) = d_j(x) = d_k(x) \leq d_h(x)\ \forall h$
$\mathcal{F}_k\big\|_{\mathcal{F}_{ij}}$	second order generalized Voronoi region	$d_i(x) = d_j(x) = d_k(x) \leq d_h(x)$

Proof of Lemma 3.13

Lemma 3.13. *When Assumption 2.12 is true, a GVG cycle cannot contain a meet point.*

Let $\mathcal{F}_{ijk}$ be the GVG edge equidistant to obstacles C_i, C_j, and C_k. Assume there is an object C_l positioned such that $x \in \mathcal{F}_{ijk}$ is a point where $d_i(x) = d_l(x)$. By Proposition 2.11, the GVG edge $\mathcal{F}_{ijl} \neq \emptyset$ and it intersects $\mathcal{F}_{ijk}$ at x. By definition of the surjective equidistant surfaces, $\mathcal{SS}_{ijk}$ and $\mathcal{SS}_{ijl}$ also intersect at x. The three-equidistant surjective surface $\mathcal{SS}_{ijl}$ must tangentially intersect $\mathcal{SS}_{ijk}$ because $\mathcal{F}_{ijk}$ is a cycle. Such an intersection is non-transversal which cannot occur when Assumption 2.12 is in effect. Therefore, there cannot be an object C_l positioned such that there is an $x \in \mathcal{F}_{ijk}$ is a point where $d_i(x) = d_l(x)$, and thus a GVG cycle cannot contain any meet points.

Proof of Lemma 3.14

Lemma 3.14. *A GVG cycle cannot contain any boundary or floating boundary points.*

Proof: Let $\mathcal{F}_{ijk}$ be a GVG edge equidistant to objects C_i, C_j, and C_k. Since the GVG edge is a cycle, it is bounded and thus, there exists a point x^* such that $D(x^*) > D(x)$ for all $x \in \mathcal{F}_{ijk}$. At x^*, $\nabla d_i(x)$ is orthogonal to $T_{x^*}\mathcal{F}_{ijk}$.

Now the proof follows by contradiction. Assume there is a point where $D(x) = 0$. That is, C_i, C_j, and C_k intersect to form a boundary point. By Lemma 3.27, introduced below, there cannot be a point where $\nabla d_i(x)$ is orthogonal to $T_x\mathcal{F}_{ijk}$. This, however, is a contradiction, and thus there cannot be a point on a GVG cycle where $D(x) = 0$.

Lemma 3.27 *If the objects $C_{i_1} \ldots, C_{i_k}$ intersect, then the associated k-equidistant surjective surface, $\mathcal{SS}_{i_1 \ldots i_k}$, is unbounded. In fact, if objects $C_{i_1}, \ldots, C_{i_k}$ intersect, then none of the gradients, $\nabla d_{i_1}(x), \ldots, \nabla d_{i_k}(x)$, is orthogonal to $T_x\mathcal{SS}_{i_1 \ldots i_k}$ for all $x \in \mathcal{SS}_{i_1 \ldots i_k}$.*

Proof: Assume that there exists an $x \in \mathcal{SS}_{i_1 \ldots i_k}$ for which $\nabla d_{i_1}(x)$ is orthogonal to $T_x\mathcal{SS}_{i_1 \ldots i_k}$. By Proposition 2.21, $\nabla d_{i_j}(x)$ must be orthogonal to $T_x\mathcal{SS}_{i_1 \ldots i_k}$ for all $j = 2, \ldots, k$ because $\nabla d_{i_1}(x)$ is orthogonal to $T_x\mathcal{SS}_{i_1 \ldots i_k}$

Let $HC_{i_1}(x)$ be the hyperplane which is orthogonal to ∇d_{i_1} and tangent to the nearest point, c_{i_1} in C_{i_1} to x. Since C_{i_1} is a convex set, C_{i_1} lies entirely on one side of $HC_{i_1}(x)$. Let $HC_{i_j}(x)$ be the hyperplanes similarly defined as above for $j = 2, \ldots, k$.

In $\mathbb{R}^m$, let S be an $(m-1)$-dimensional sphere centered at x with radius $D(x)$. By Assumption 2.12, the k closest points on the k closest obstacles form a $(k-1)$-dimensional hyperplane orthogonal to $T_x\mathcal{SS}_{i_1 \ldots i_k}$ and passing through x. For $k > 2$, these k points define a $(k-2)$-dimensional sphere, termed a *sub-sphere* (S_{sub}) which is a subset of S and has a radius less than or equal to $D(x)$. When the radius of S_{sub} is equal to the radius of S, we say that S_{sub} is a *major sub-sphere* of S.

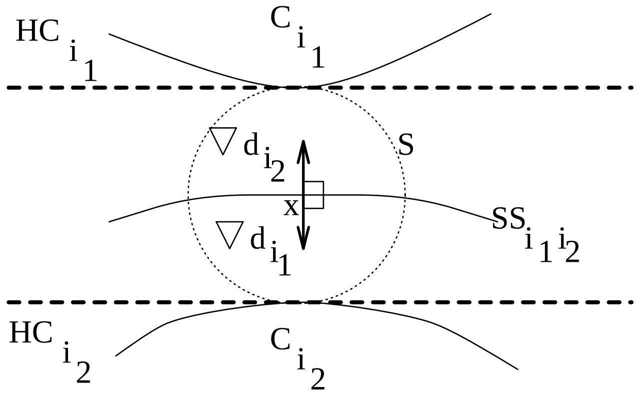

Figure 3.20 The circle represents the largest sphere, centered at x, that does not penetrate any obstacle. Since x is a point on a two-equidistant surjective surface, $\mathcal{SS}_{i_1 i_2}$, only two obstacles are tangent to S. The separating planes for these objects are delineated by thick dotted lines.

For $j = 1, \ldots, k$, HC_{i_j} is tangent to S and passes through a point on the $(k-2)$-dimensional sub-sphere, S_{sub}.

Next, it needs to be shown that $HC_{i_1}, \ldots, HC_{i_k}$ cannot intersect at a point. We first show this when $k = 2$. The sphere S has co-dimension one. If ∇d_{i_1} and ∇d_{i_2} are orthogonal to $T_x \mathcal{SS}_{i_1 i_2}$, then there are only two points (at opposite poles of the sphere) where the separating planes are tangent to S. In this case, the separating planes are parallel to each other (Figure 3.20). By definition of a convex set, if the separating planes never intersect, then the obstacles cannot intersect. This is a contradiction. Therefore, for $k = 2$, no gradient vector can be orthogonal to $T_x \mathcal{SS}_{i_1 \ldots i_k}$, and thus $\partial D(x)$ has no local maxima. Since $\partial D(x)$ has no local maxima, it has no global maxima and thus $\mathcal{SS}_{i_1 \ldots i_k}$ is unbounded.

In general, if the gradients $\nabla d_{i_1}(x), \ldots, \nabla d_{i_k}(x)$ are orthogonal to $T_x \mathcal{SS}_{i_1 \ldots i_k}$, then S_{sub} and S have the same radius. In other words, S_{sub} is a major sub-sphere. It can be shown that if k hyperplanes are tangent to S at a point in S_{sub}, then they can never intersect at a point. Since $HC_{i_1}, \ldots, HC_{i_k}$ can never intersect at a point, the obstacles $C_{i_1}, \ldots, C_{i_k}$ cannot intersect at a point either. This is a contradiction, and thus $\mathcal{SS}_{i_1 \ldots i_k}$ is unbounded. ▼

A similar argument can be made for floating boundary points. ■

Proof of Lemma 3.15

Lemma 3.15. *In $\mathbb{R}^3$, a three-equidistant surface, $\mathcal{SS}_{ijk}$, is either diffeomorphic to S^1 (i.e., it is a GVG cycle), or it is unbounded.*

Proof: First, we show the case when C_i, C_j, and C_k are disjoint convex sets and then we show the case when they overlap. A two-equidistant surjective surface for two disjoint convex sets is a separator in $\mathcal{W}$ by the following lemma.

Lemma 3.28 *A two-equidistant surjective surface for two disjoint convex sets is an unbounded separator in $\mathcal{W}$.*

Proof: Let $\mathcal{SS}_{ij}$ be a two equidistant surface for obstacles C_i and C_j in the space $\mathcal{W}\backslash(\bigcup_{h\neq i,j} C_h)$. Let $\mathcal{S}_i$ be the set of points in $\mathcal{W}\backslash(\bigcup_{h\neq i,j} C_h)$ closer to C_i than C_j. Define $\mathcal{S}_j$ similarly. That is,

$$\mathcal{S}_i = \{x \in \mathcal{W}\backslash(\bigcup_{h\neq i,j} C_h) : d_i(x) \leq d_j(x)\}$$

$$\mathcal{S}_j = \{x \in \mathcal{W}\backslash(\bigcup_{h\neq i,j} C_h) : d_j(x) \leq d_i(x)\}. \quad (3.10)$$

The two-equidistant surjective surface, $\mathcal{SS}_{ij}$, is on the boundary of both $\mathcal{S}_i$ and $\mathcal{S}_j$.

Let F_i^t be the flow of ∇d_i and let $\mathcal{SS}_i = \{x \in \mathcal{S}_i : F_i^t \in \mathcal{SS}_{ij} \text{ for some } t\}$. Define $\mathcal{SS}_j$ similarly.

The mapping $F_i\colon \mathbb{R}\times\mathcal{S}_i \to \mathcal{W}\backslash C_i$ is a continuous mapping. Let $Q_i = F^{-1}(\mathcal{SS}_{ij})$. Q_i is closed because it is the pre-image of a closed set under a continuous mapping. Note that $(t,x) \in Q_i$ if and only if $F_i^t(x) \in \mathcal{SS}_{ij}$. (So, $x \in \mathcal{SS}_{ij}$ implies that $(0,x) \in Q_i$.

Since $\mathcal{SS}_{ij}$ is the image of a connected set under a continuous mapping F_i, $\mathcal{SS}_{ij}$ is connected. Therefore, $\mathcal{S}_i$ and $\mathcal{S}_j$ share a common connected boundary: $\mathcal{SS}_{ij}$. This boundary separates $\mathcal{W}$ into two regions: points closer to C_i and points closer to C_j. Since $\mathcal{S}_i$ and $\mathcal{S}_j$ are both unbounded, so must be $\mathcal{SS}_{ij}$.

▼

By a similar argument, it can be shown that $\mathcal{SS}_{ijk}$ is a separator on $\mathcal{SS}_{ij}$. In $\mathbb{R}^3$ the pre-image theorem asserts that $\mathcal{SS}_{ijk}$ is one-dimensional. By the Jordan curve lemma, $\mathcal{SS}_{ijk}$ in $\mathbb{R}^3$ can either be (1) a manifold diffeomorphic to S^1 or (2) an unbounded manifold diffeomorphic to $\mathbb{R}$.

Now, we consider the case where the obstacles, C_i, C_j, and C_k, intersect to form a boundary point, C_{ijk}, out of which $\mathcal{SS}_{ijk}$ emanates. Lemma 3.27 asserts that $\mathcal{SS}_{ijk}$ is unbounded. ■

Proof of Lemma 3.16

Lemma 3.16. A GVG^2 equidistant edge can only intersect the GVG at a meet point.

Proof: Consider the GVG^2 equidistant edge $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$. For all points $x \in \mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$, $d_k(x) = d_l(x) \geq d_i(x) = d_j(x)$. When $d_i(x) = d_j(x) = d_k(x)$ for some $x \in \mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$, $d_i(x) = d_j(x) = d_k(x) = d_l(x)$. However, x cannot be in the interior of a generalized Voronoi edge $\mathcal{F}_{ijk}$ because for all $y \in \mathrm{int}(\mathcal{F}_{ijk})$, $d_i(y) = d_j(y) = d_k(y) < d_h(y)$ for all h. ■

Proof of Lemma 3.18

Lemma 3.18. *If the three-equidistant face $\mathcal{F}_{ijk}$ is not empty, then the second order generalized Voronoi region $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ must not be empty. Furthermore, if $\mathcal{F}_{ijk} \neq \emptyset$, then $\mathcal{F}_{ijk} \subset \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$.*

Proof: Pick $x \in \mathrm{int}(\mathcal{F}_{ijk})$. By definition, $d_i(x) = d_j(x) = d_k(x) < d_h(x)$ for all $h \neq i, j, k$. For a nbhd(x), let $Y(x) = \mathrm{nbhd}(x) \bigcap (\mathrm{int}(\mathcal{F}_{ij}) \setminus \mathrm{int}(\mathcal{F}_{ijk}))$. By definition, $d_i(y) = d_j(y) < d_k(y)$ for all $y \in Y(x)$. By continuity of the single object distance function, for all $h \neq i, j, k$, $d_k(y) < d_h(y)$ for all $y \in Y(x)$. Therefore, for all $y \in Y(x)$, $d_i(y) = d_j(y) < d_k(y) < d_h(y)$ for all $h \neq i, j, k$.

Therefore, there exists a region where C_i and C_j are the closest obstacles and C_k is the second closest. That is, $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}} \neq \emptyset$. Furthermore, $\mathcal{F}_{ijk}$ is a subset of $\partial\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$

because:

$$\begin{aligned}\mathcal{F}_{ijk} &= \{x : \forall h\, d_h(x) \geq d_i(x) = d_j(x) = d_k(x)\} \\ &\subset \partial\{x : \forall h\, d(h) \geq d_k(x) \geq d_j(x) = d_i(x)\} \\ &= \partial \mathcal{F}_k\Big|_{\mathcal{F}_{ij}},\end{aligned}$$

where ∂ is the boundary operator.

Therefore, by definition, if $\mathcal{F}_{ijk} \neq \emptyset$, then it is a non-empty subset of the boundary of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. ■

Proof of Lemma 3.19.

Lemma 3.19. The boundary of a second order generalized Voronoi region contains at most *one* three-equidistant face. That is, $\mathcal{F}_{pqr} \subsetneq \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ for all $\{p, q, r\} \neq \{i, j, k\}$.

Proof: Assume that $\mathcal{F}_{ijk}$ and $\mathcal{F}_{ijl}$ are on the boundary of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. By definition, for all $x \in \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$, it must be true that $d_l(x) \geq d_k(x) \geq d_i(x) = d_j(x)$ for all $l \neq i, j, k$. Since $\mathcal{F}_{ijl} \subset \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ (by assumption), for all $x \in \mathcal{F}_{ijl}$, $d_k(x) \leq d_l(x)$. Thus, for all $x \in \mathcal{F}_{ijl}\backslash\mathcal{F}_{ijk}$ (which is a subset of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$), $d_k(x) \leq d_l(x)$. However, this is a contradiction because for all $x \in \mathcal{F}_{ijl}\backslash\mathcal{F}_{ijk}$, $d_l(x) < d_k(x)$. ■

3.3 Extended Boundedness Assumption

The *Extended Boundedness Assumption* (formally defined in Section 3.3.1) guarantees that there are no HGVG cycles and outer periods (Section 3.3.3). This assumption leads to the basis for deriving important results necessary to prove connectivity of the HGVG. In particular, in Section 3.3.2, we show that all second order generalized Voronoi regions contain a generalized Voronoi edge when the Extended Boundedness Assumption is true. The following example motivates the need for this assumption.

Example 3.29 Recall the environment from Example 3.1 that consists of a room with a box in its interior. See Figure 3.1. In Figure 3.2, it can be seen the two-

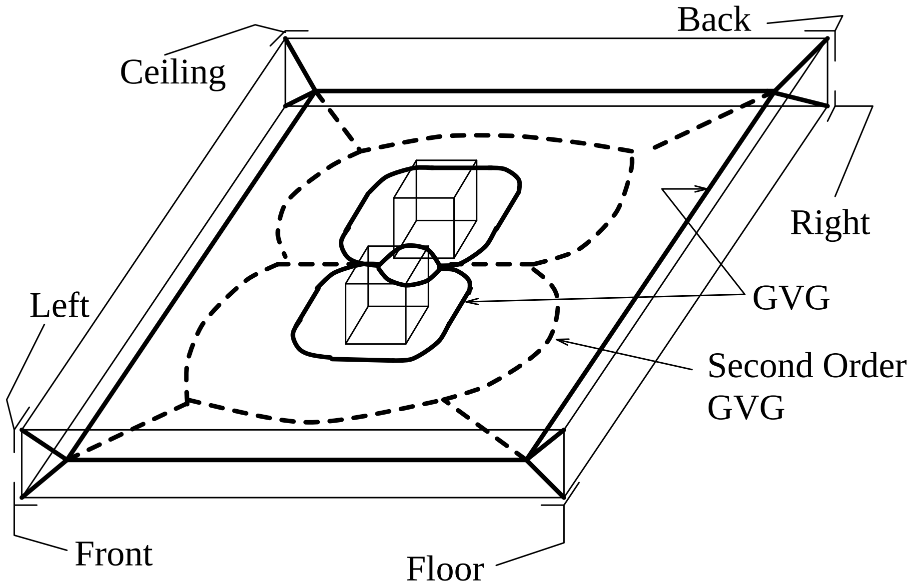

Figure 3.21 Room with two boxes in its interior. The solid lines are GVG edges and the dotted lines are GVG^2 edges.

equidistant face defined by the floor and ceiling has a disconnected boundary and thus the HGVG is also disconnected.

However, when an additional box is placed in the middle of the room, the HGVG on two-equidistant face defined by the floor and ceiling *becomes connected.* See Figures 3.21 and 3.22. ♦

3.3.1 Definition

Assumption 3.30 (Extended Boundedness) In $\mathbb{R}^m$, each p-order k-equidistant face has at least one p-order $(k+1)$-equidistant face on its boundary.

In $\mathbb{R}^3$ $(m = 3)$ this assumption implies that all generalized Voronoi edges ($k = 3$, $p = 1$) contain at least one meet point. That is, for all i, j, k, there exists $x \in \mathcal{F}_{ijk}$ and there exists an l, such that $d_l(x) = d_k(x)$. By the Equidistant Surface Transversality Assumption (Assumption 2.12), this point is isolated.

In Example 3.29, the environment in Figure 3.1 contains a generalized Voronoi edge which has no meet points; it is the cycle which surrounds the inner box. Nev-

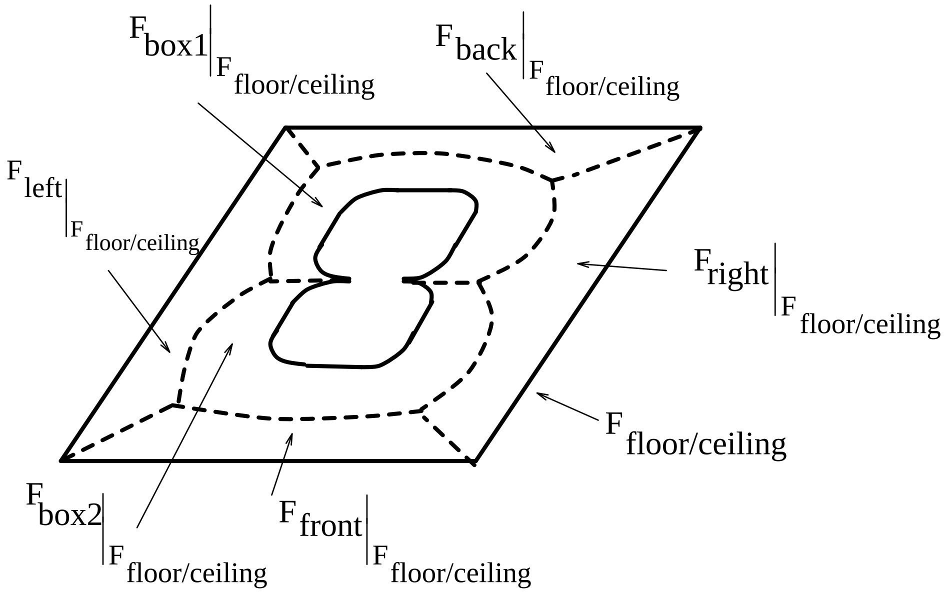

Figure 3.22 The HGVG on the two-equidistant face, $\mathcal{F}_{floor/ceiling}$, is now connected.

ertheless, when an additional box enters the environment (Figure 3.21), additional meet points, associated with the inner box, appear, thereby forming a connected HGVG. In this scenario, the Extended Boundedness Assumption holds.

The purpose of this section is to show how Assumption 3.30 leads to cycle-free environments. First, it is shown in Section 3.3.2 that under the Extended Boundedness Assumption, all visible second order generalized Voronoi regions contain a generalized Voronoi edge. Using this result, it is proven in Section 3.3.3 that environments satisfying the Extended Boundedness Assumption are cycle-free.

Note that this assumption requires use of the visible (or slanted) distance function. That is, the robot is only aware of obstacles that are within line of sight of it. Recall, that all structures are defined in terms of the visible distance function. Also note that when this assumption is upheld, all second order generalized Voronoi edges ($k = 2$, $p = 2$) have at least one second order meet point.

3.3.2 Second Order Generalized Voronoi Regions

In order to guarantee the connectivity of the HGVG, we need to show that the boundary of each second order generalized Voronoi region is connected. In order to do this, first we must show that every second order generalized Voronoi region contains a second order generalized Voronoi edge, as demonstrated by the following lemma. Recall that in $\mathbb{R}^3$, a three-equidistant face is a GVG edge.

Lemma 3.31 *Let the Extended Boundedness Assumption (Assumption 3.30) and the visible distance function be in effect. In this case, all second order generalized Voronoi regions must contain a three-equidistant face.*

Proof: Recall the definition of the second order generalized Voronoi region,

$$\mathcal{F}_k\Big|_{\mathcal{F}_{ij}} = \{x \in \mathcal{F}_{ij} : \forall h \notin \{i,j,k\}\, d_h(x) \geq d_k(x) = d_i(x)\}.$$

Given the Extended Boundedness Assumption (Assumption 3.30), there exists an $h' \notin \{i,j\}$ and an x such that $d_i(x) = d_j(x) = d_{h'}(x)$. If $h' = k$, then $\mathcal{F}_{ijk} \neq \emptyset$, and by Lemma 3.18 and Lemma 3.19, it is the only three-equidistant face in $\partial\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$.

If $h' \neq k$, then that implies $\mathcal{F}_{ijh'}$ must exist (i.e., there exists an x such that $d_i(x) = d_j(x) = d_{h'}(x)$). However, since the second order generalized Voronoi region $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}} \neq \emptyset$, it must be true that $d_k(y) \leq d_{h'}(y)$ for all $y \in \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. By continuity of the single object distance function, $\mathcal{F}_{ijk}$ must also be a non-empty subset of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ (Lemma 3.18). This is a contradiction of Lemma 3.19, where only one three-equidistant face may be a subset of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. Therefore, $h' = k$, and $\mathcal{F}_{ijk}$ is always a subset of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. ■

Example 3.32 The Extended Boundedness Assumption is based solely on the use of line of sight information because it is stated in terms of the visible (or slanted) distance function. In this example, a second order generalized Voronoi region, defined in terms of the X-distance function, may exist without containing a GVG edge.

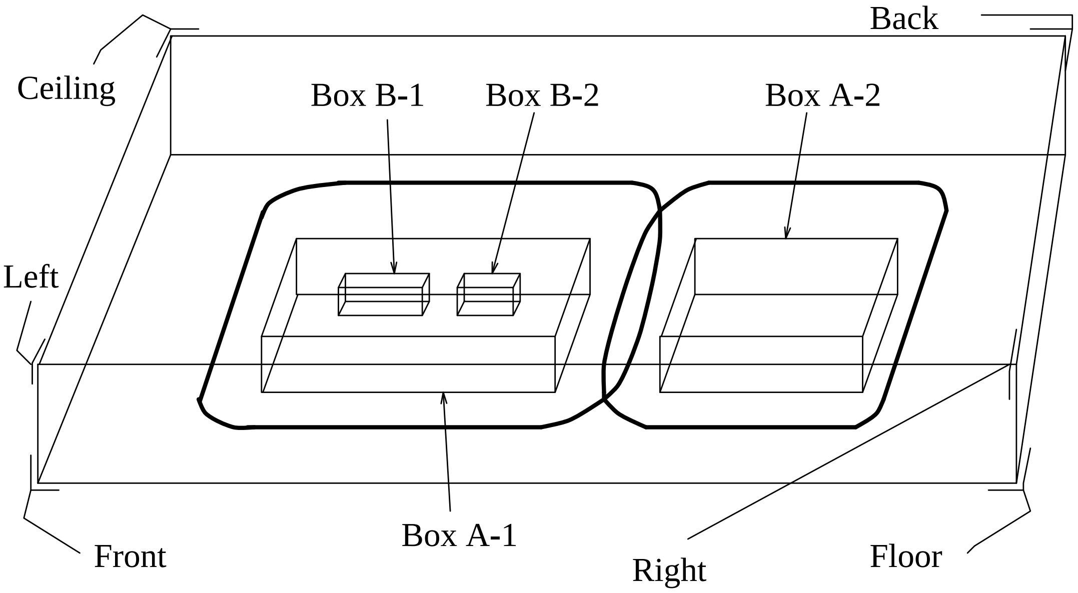

Figure 3.23 A room with four boxes in its interior. Two of the boxes are on top of the box on the left box. The thick solid lines represent the GVG edges associated with the floor, ceiling, box $A-1$, and box $A-2$.

However, such a region would not exist for one defined in terms of the visible (or slanted) distance function.

Figure 3.23 contains the same room as in Example 3.29, but there are two more boxes on top of one of the boxes. Figure 3.24 is a view of the environment from below. From this perspective, obstacle A-1 occludes obstacles B-1 and B-2. The occluded obstacles are represented as dotted lines. The solid lines signify the GVG edges.

The dotted lines in Figures 3.25 and 3.26 delineate the GVG^2 edges, *defined in terms of the X-distance function.* These GVG^2 edges form an inner GVG^2 period in the second order generalized Voronoi region in Figure 3.26. It is defined in terms of obstacles, B-1 and B-2, which are not within line of sight of the GVG edges. This period contains no GVG edge in its interior, which seemingly contradicts Lemma 3.31. However, Lemma 3.31 makes use of the visible distance function, which precludes the existence of this period in this example. ⧫

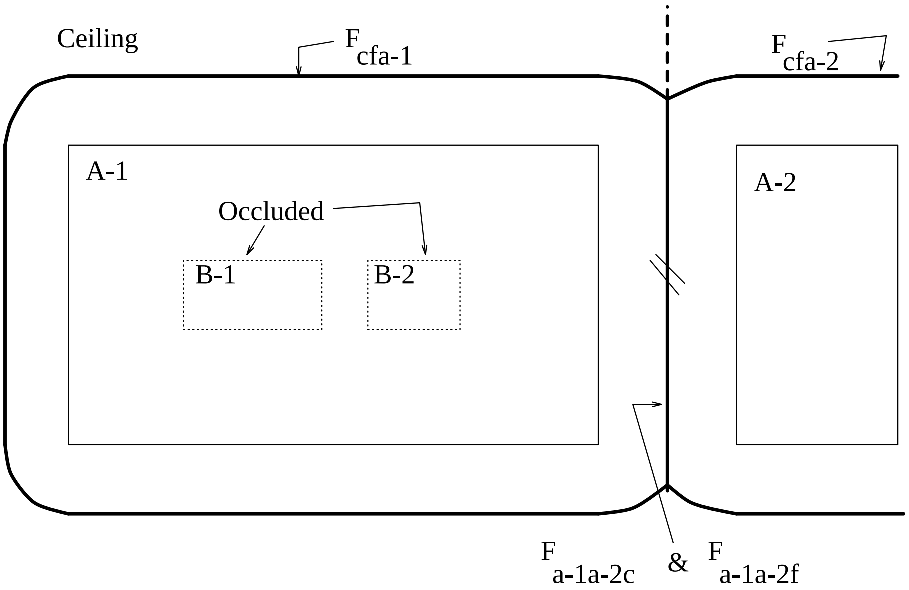

Figure 3.24 Bottom view of a portion of the environment shown in Figure 3.23. The dotted lines signify that objects B-1 and B-2 are occluded by object A-1. The solid lines represent GVG edges. The two strikes on the vertical solid line signify there are two GVG edges, but one occludes the other. The ceiling is into the page.

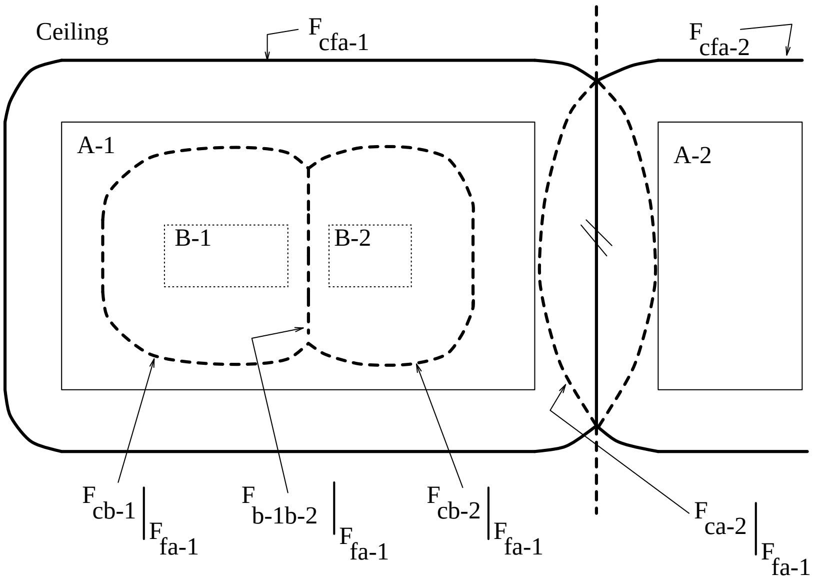

Figure 3.25 Same as Figure 3.24, except the GVG^2 edges are drawn as dashed lines.

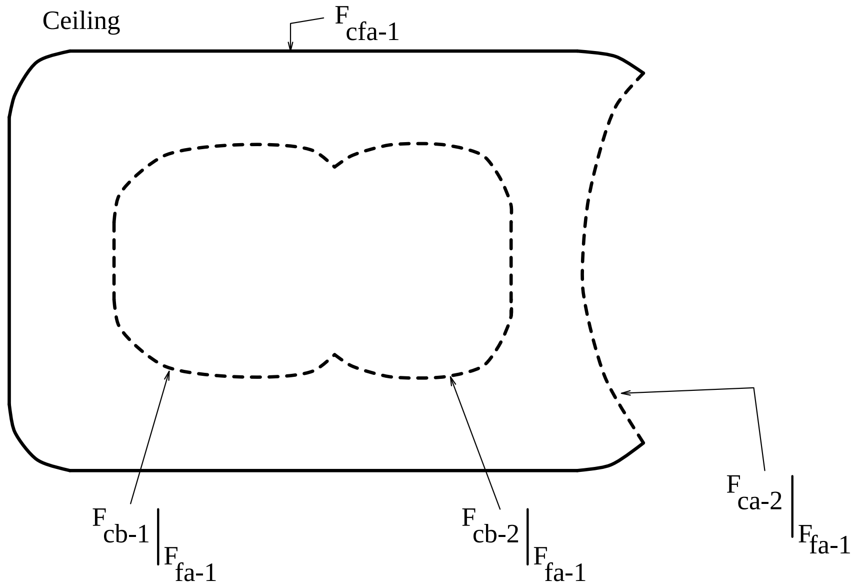

Figure 3.26 Second order generalized Voronoi region, $\mathcal{F}_{ceiling}\big|_{\mathcal{F}_{floor/a-1}}$, with inner period and no GVG edge.

3.3.3 Cycle-free Environment

In $\mathbb{R}^m$ when $m > 2$, the Extended Boundedness Assumption is true for most "cluttered" workspaces. Robots whose configuration spaces are high dimensional tend to be highly articulated and are thus better suited for cluttered environments. Such environments do not contain cycles and thus may contain a connected HGVG. The following lemma states how the Extended Boundedness Assumption leads to HGVG's that are free of cycles and which are therefore connected.

Lemma 3.33 *If Assumptions 2.12 and 3.30 hold, then there will be no GVG cycles, no GVG*^{2} *cycles, and no outer GVG*^{2} *periods.*

Proof: Let $\mathcal{F}_{ijk}$ be a generalized Voronoi edge in $\mathbb{R}^3$. By the Extended Boundedness Assumption (Assumption 3.30) there exists a point $x \in \mathcal{F}_{ijk}$ such that there is an obstacle C_l that is positioned such that $d_l(x) = d_k(x)$. Therefore, $\mathcal{F}_{ijkl} = \mathcal{F}_{ijl} \bigcap \mathcal{F}_{ijk} \neq \emptyset$. Since $\mathcal{F}_{ijk}$ is not disconnected from all other GVG edges, when the Equidistant Surface Transversality Assumption (Assumption 2.12) is in effect, Proposition 3.12 asserts that $\mathcal{F}_{ijk}$ is not a cycle.

By Proposition 3.26, if there exists (1) an outer second order period, which is a component of the boundary of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$, and (2) a generalized Voronoi edge which is a subset of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ (whose existence is guaranteed by the Lemma 3.33), then there exists a first order cycle. The contrapositive of this statement is also true. If a GVG cycle does not exist, then an outer GVG^2 period cannot exist *or* the Extended Boundedness Assumption is not valid.

The Extended Boundedness Assumption implies that a GVG cycle cannot exist. This implies that an outer GVG^2 period cannot exist *or* the Extended Boundedness Assumption is not in effect. However, since the Extended Boundedness Assumption is in effect, there cannot be any outer GVG^2 periods. ▼

The following chart illustrates the flow of logic for the last two paragraphs of the previous proof.

$$\boxed{\text{Extended Boundedness Assumption}} \Longrightarrow \boxed{\text{no GVG cycle}} \quad \textit{and} \quad \boxed{\text{GVG edge exists in second order generalized Voronoi region}}$$

$$\boxed{\text{GVG edge exists in second order generalized Voronoi region}} \quad \textit{and} \quad \boxed{\text{outer second order period}} \Longrightarrow \boxed{\text{GVG cycle}}$$

$$\boxed{\text{no GVG cycle}} \Longrightarrow \boxed{\text{no outer second order period}} \quad \textit{or} \quad \boxed{\text{no GVG edge exists in second order generalized Voronoi region}}$$

$$\boxed{\text{Extended Boundedness Assumption}} \Longrightarrow \boxed{\text{no GVG cycle}} \Longrightarrow \boxed{\text{no outer second order period}}$$

3.3.4 The Extended Boundedness Assumption and Planar Environments

There already exists a large volume of sensor based path planning strategies which only work in the plane. It is our belief that barring all sensor-related issues (e.g., sensor error, sensor fusion, dead-reckoning, etc.), path planning in the plane is trivial. In this section, we show that the Extended Boundedness Assumption is always true in the planar case. This *could* be a reason as to why path planning in the plane is much easier than in higher dimensions.

Lemma 3.34 *In all bounded environments, when $k = 2$, the Extended Boundedness Assumption (Assumption 3.30 is always valid.*

Proof: This is a consequence of Proposition C.9 which states that the generalized Voronoi diagram is connected. Recall that the generalized Voronoi diagram is the union of all two-equidistant faces (generalized Voronoi faces). The Boundedness Assumptions (Assumption 1.1) guarantees that there will be more than one two-equidistant face. If there were only two disjoint obstacles, then there would be only one (unbounded) two-equidistant face. However, in the planar case the Boundedness Assumption guarantees that there must be at least three obstacles, which form a triangle.

By Proposition C.9 the union of the two-equidistant faces is a connected set and by Proposition 2.11, the faces are connected through a point that is equidistant to three obstacles (a three-equidistant face). That is, for every pair of obstacles, there always exists a third obstacle that is positioned so that there is one point which is equidistant to all three obstacles. ■

Corollary 3.35 There cannot be any cycles in planar environments.

Proof: Lemma 3.34 implies that the Extended Boundedness Assumption is always true in the planar case, and when the Extended Boundedness Assumption is made, by Lemma 3.33 there cannot be any GVG cycles. Therefore, there cannot be any GVG cycles in planar environments. ■

3.4 Connectivity

Recall the claim from Section 3.1 that the connectivity problem has been reduced to showing that each boundary of a second order generalized Voronoi region is connected (or can be readily connected with a link). This claim is valid because the $\mathrm{GVG}^2\Big|_{\mathcal{F}_{ij}}$ is a generalized Voronoi complex (Section 3.4.1). Guaranteeing connectivity of the second order generalized Voronoi regions (and thus the HGVG) with line of sight information relies on the introduction of some additional links, described in Section 3.4.2. With these links, the boundary of each of the second order generalized Voronoi regions is connected (Section 3.4.3) and thus the HGVG, with these links, (Section 3.4.4) is connected. We term the HGVG with its links the *linked hierarchical generalized Voronoi graph (L^3HGVG).*

3.4.1 The HGVG and the Generalized Voronoi Complex

Connectivity of the HGVG on a two-equidistant face will be shown by exploiting properties of the *generalized Voronoi complex,* defined in Section C.2. Since the generalized Voronoi complex is a connected roadmap (Section C.2), the issue of L^3HGVG connectivity can be reduced to showing that the L^3HGVG on a two-equidistant face forms a generalized Voronoi complex.

The generalized Voronoi complex is composed of *generalized Voronoi cells*, which are defined in Section C.2. In order to show that a L^3HGVG is a generalized Voronoi complex, we must first demonstrate that each second order generalized Voronoi region, $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$, is a generalized Voronoi cell. That is, the following properties must hold:

1. $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ is a connected set,
2. $\dot{\partial}\mathcal{F}_k\Big|_{\mathcal{F}_{ij}} \neq \emptyset$,
3. $\dot{\partial}\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ is a connected set,

where $\dot{\partial}$ is the "sub-boundary" operator defined in Section C.2. The sub-boundary of a set is the portion of the set boundary that is contained in the set. For closed sets, the sub-boundary is the same as the set's boundary.

If $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ has many connected components, then each connected component is treated as a separate set. Thus, the first property is trivially met. The Extended Boundedness Assumption (Assumption 3.30) insures that the second property is easily met because Assumption 3.30 guarantees that for each second order generalized Voronoi region, there exists a generalized Voronoi edge on its boundary. By Lemma 3.18, this edge is a subset of the boundary of the second order generalized Voronoi region. The third property, which states that $\dot{\partial}\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ is a connected set, is demonstrated in Section 3.4.3.

Once it can be shown that the boundary of all second order generalized Voronoi regions are connected, then the second order generalized Voronoi regions form the basis of a generalized Voronoi complex. Therefore, the L^3HGVG on each two-equidistant face is connected, and thus the L^3HGVG is connected.

3.4.2 Links

The L^3HGVG's connectivity hinges on showing that the boundary of each second order generalized Voronoi region is connected, which in general is not the case. Even when the Extended Boundedness Assumption is valid, second order generalized Voronoi regions could contain disconnected boundaries, which give rise to disconnected HGVG's. Additional structures, termed *links*, are used to connect disconnected boundary components. In this section, we introduce all of the links that are necessary to guarantee connectivity of the L^3HGVG.

The Extended Boundedness Assumption (Assumption 3.30) guarantees that there will be no GVG and GVG^2 cycles. As a result of Lemma 3.33, the extended boundedness assumption also guarantees there cannot be any outer GVG^2 periods. However, this does not preclude the possibility of the existence of an inner GVG^2 period. There are three types of inner periods: (1) one formed by two or more GVG^2 equidistant edges, zero or more boundary edge fragments, and zero or more occluding edge fragments, (2) an occluding period, and (3) a boundary period. The following lemmas show that a link to each of these inner GVG^2 periods exists by construction. The proofs of these Lemmas are in Section 3.4.5. The implementation of such a linking procedure provides the desired connectivity.

Example 3.36 (Inner Period) Figure 3.27 contains an environment similar to that of Example 3.32, where the existence of an inner period was dependent upon the choice of distance function. In Example 3.32, the HGVG, defined in terms of the X-distance function, contains an inner second order period. However, when the V-distance function is used to define the HGVG, the inner period disappeared. In this example, we show that an inner second order period can exist using the V-distance

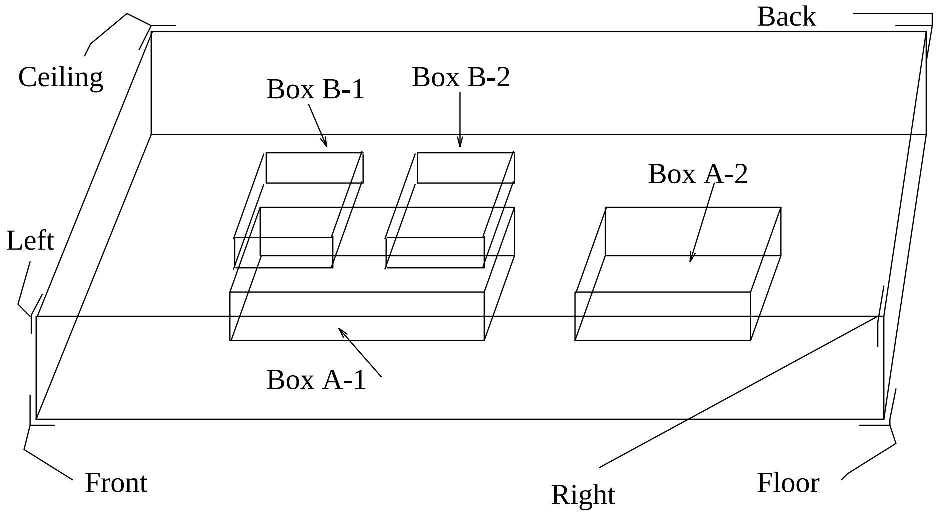

Figure 3.27 A room with four boxes floating in its interior. Boxes B - 1 and B - 2 are floating above box A and have the same depth as box A.

function; that is, even when line of sight information is exclusively used, an inner second order period can exist.

Figure 3.27 contains a room with four boxes floating in its interior. Two of the boxes, objects B - 1 and B - 2, are above box A - 1. The environment in Figure 3.27 differs from the environment in Example 3.32 in that boxes B - 1 and B - 2 have the same depth as box A - 1.

Figure 3.28 depicts a cross-section of a three-dimensional world depicted in Figure 3.27. The cross-sections of the two-equidistant faces are drawn as solid lines and arc segments. The cross-sections of the GVG edges are points where three edges intersect and have circles drawn around them. Figures 3.29 and 3.30 display a top view of Figure 3.27. In these figures, the solid lines are the GVG edges and the dotted lines are the GVG^2 edges. In Figure 3.30 it can be seen that the second order generalized Voronoi region has an outer and inner boundary. Lemma 3.37 allows for a link to be made between the two boundaries. ♦

Lemma 3.37 (Inner Boundary Link) *If an inner GVG^2 period with GVG^2 edges exists on the boundary of the second order generalized Voronoi region, then a link exists from the outer boundary to it.*

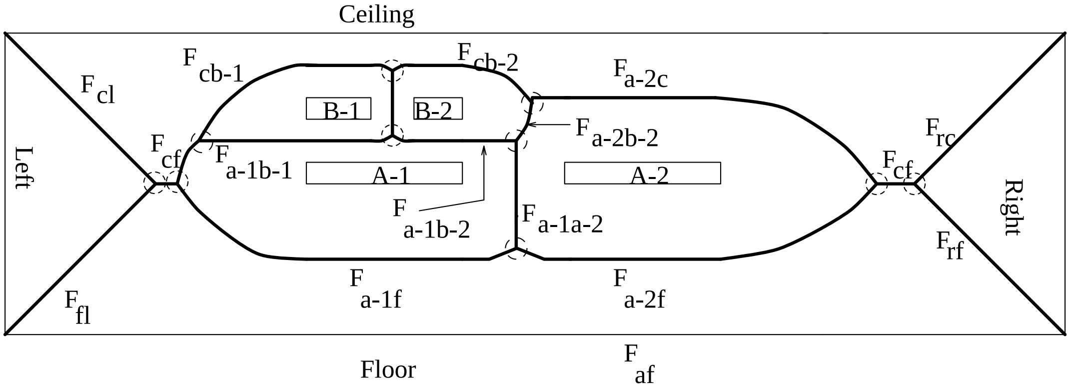

Figure 3.28 Cross-section of the environment in Figure 3.27. The cross-section is parallel to the front face of the rectangular enclosure and cuts it through the three floating boxes. The solid lines are the two-equidistant faces which meet at generalized Voronoi edges, which are circled.

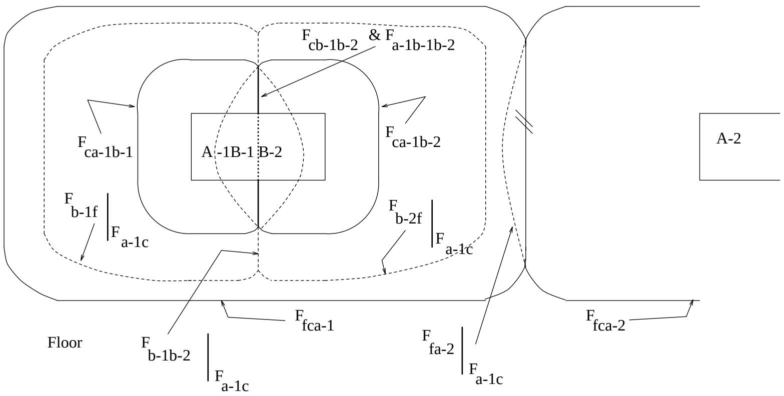

Figure 3.29 GVG and GVG^2 edges (Top View).

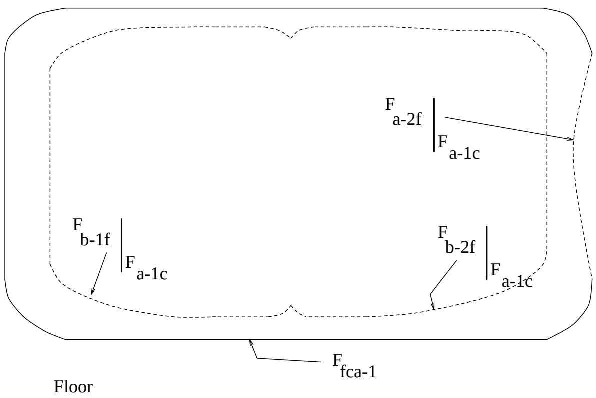

Figure 3.30 Inner Period.

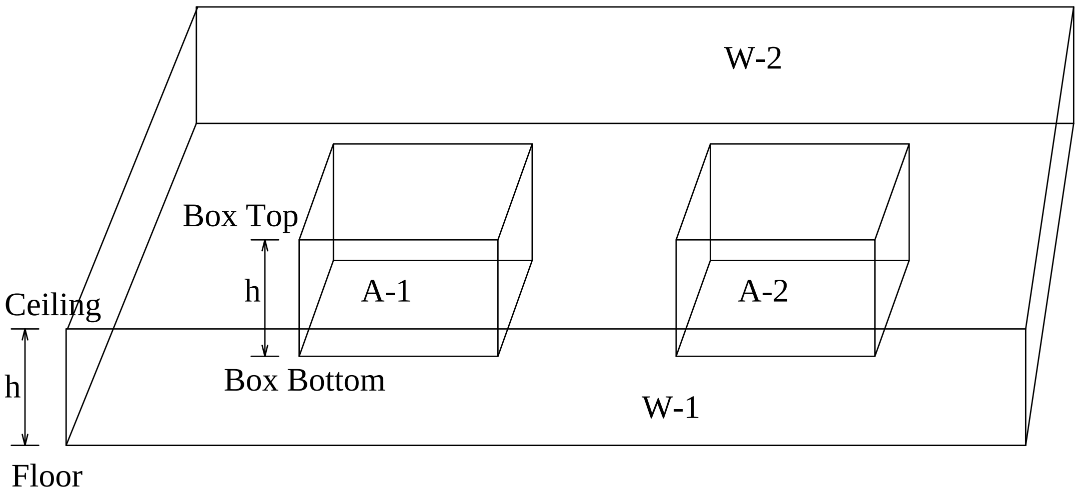

Figure 3.31 Two boxes in a room.

The proof of this lemma appears in Section 3.4.5.

The two remaining cases deal with an occluding period and a boundary period. These cases share a common link creation strategy: gradient ascent or descent of the multi-object distance function constrained to a two-equidistant face.

Example 3.38 (Boundary Period) Figure 3.31 depicts a room with two boxes in its interior such that the height of the boxes is the same as the height of the room (the ceiling and floor intersect the top and bottom of the box, respectively).

The existence of a boundary period is dependent upon the relative distance between the interior boxes. Figures 3.32 and 3.33 depict the two possible top views of the environment shown in Figure 3.31. In Figure 3.32, the boxes are far enough apart for a boundary period to exist. The dashed lines which trace out the top of boxes A - 1 and A - 2 represent the boundary periods. The dotted lines represent the GVG^2 edges, and do not interact with the boundary period at all. The solid lines are the GVG edges. In Figure 3.33, the boxes are close enough to each other to preclude the existence of the boundary periods. In this figure, the GVG^2 edges intersect the boundary fragments. For the environment which appears in Figure 3.32, it is necessary to construct a link to the boundary period in order to ensure connectivity of the HGVG. Lemma 3.39 ensures the existence of such a link.

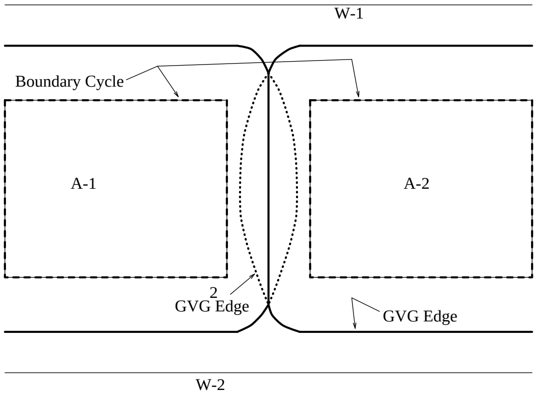

Figure 3.32 Top View of Figure 3.31 when the boxes are far apart. The dashed lines delineate the two boundary periods, one defined by the ceiling and box A-1, and the other defined by the ceiling and box A-2.

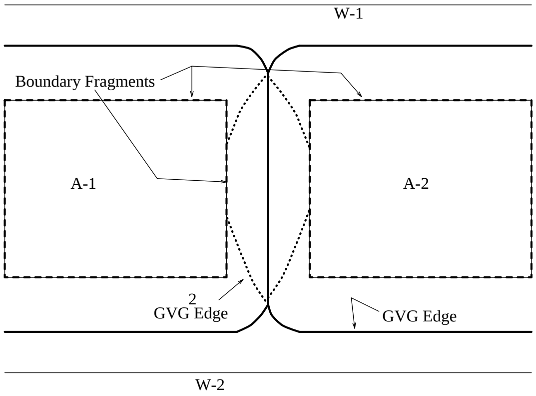

Figure 3.33 Top View of Figure 3.31 when the boxes are not far apart. The dashed lines delineate boundary fragments.

♦

Lemma 3.39 (Boundary Link) *In $\mathbb{R}^3$, if a boundary period exists on the boundary of the second order generalized Voronoi region, then it must be an "inner boundary" and a link exists from the outer boundary to it.*

The proof of this lemma appears in Section 3.4.5.

Example 3.40 (Occluding Link) Disconnected boundaries also arise with occluding periods and, in fact, are quite similar to disconnectivities found with boundary periods, described in the previous example.

The example in Figure 3.34 is quite similar to the one found in Figure 3.31 from the previous example. In Figure 3.34, the boxes in the center of the room are not as high as the ceiling. As was the case in the previous example, the existence of occluding periods depends on the relative distance between the two interior boxes. Figures 3.35 and 3.36 depict the two possible top views of the environment shown in Figure 3.34. In Figure 3.35, the boxes are far enough apart for an occluding period to exist. The dashed lines which trace out the top of boxes A - 1 and A - 2

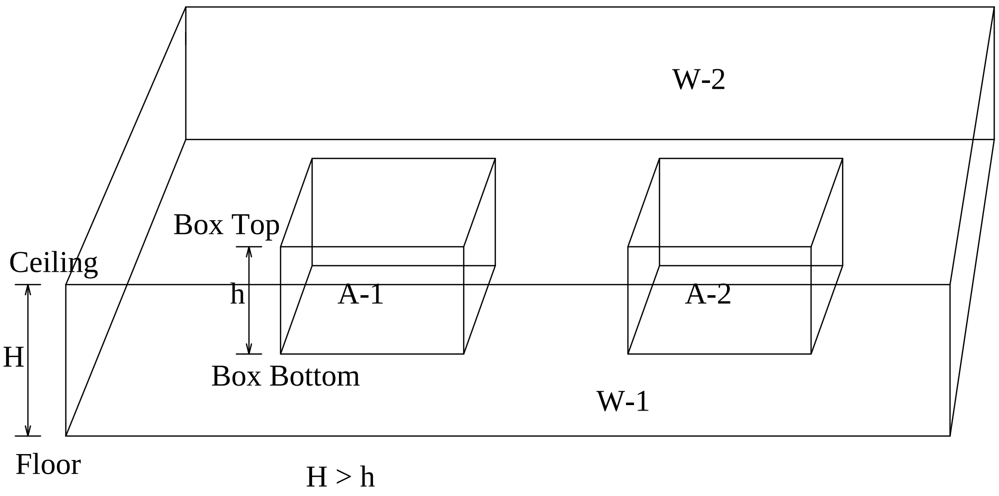

Figure 3.34 Two boxes in a room where the height of the boxes is less than the height of the room.

represent the occluding periods. The dotted lines represent the GVG^2 edges, and do not interact with the occluding period at all. The solid lines are the GVG edges. In Figure 3.36, the boxes are close enough to each other to preclude the existence of the occluding periods. In this figure, the GVG^2 edges intersect the occluding fragments. In Figure 3.35, it is necessary to construct a link to the occluding period to ensure connectivity of the HGVG. Lemma 3.41 ensures the existence of such a link. ♦

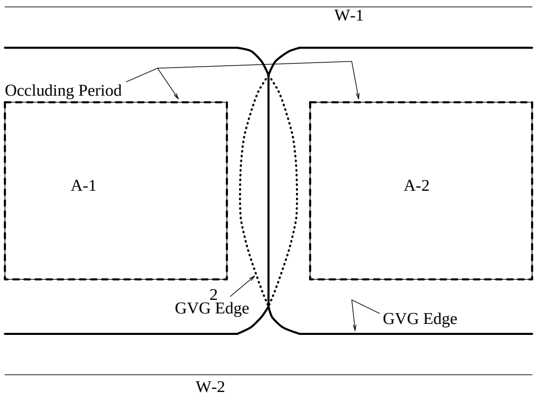

Figure 3.35 Occluding Period.

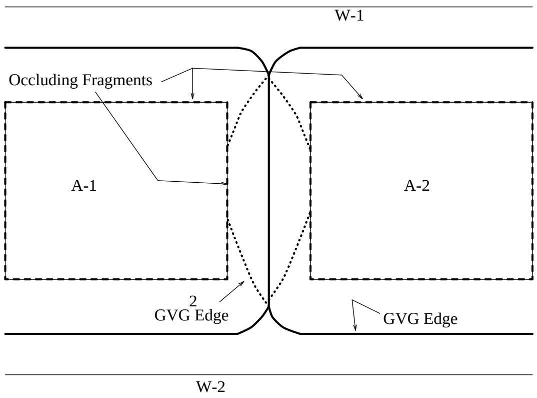

Figure 3.36 Occluding Fragment.

Lemma 3.41 (Occluding Link) *In $\mathbb{R}^3$, if an occluding period exists on the boundary of a second order generalized Voronoi region, then a link can be made to the period.*

The proof of this lemma appears in Section 3.4.5.

The structure that was implicitly defined via the linking procedure described above is called the L^3HGVG, which is the *linked hierarchical generalized Voronoi graph.* The L is cubed because there are three types of links: one for inner GVG^2 periods, one for boundary periods, and one for occluding periods.

3.4.3 Connected Boundary with Links

The final step in showing that second order generalized Voronoi regions are generalized Voronoi cells is to show their boundaries are connected (with the possible use of links). We term the second order generalized Voronoi regions with links, *linked second order generalized Voronoi regions.* The following theorem will be useful in proving the connectivity of the L^3HGVG.

Proposition 3.42 *Given the Boundedness Assumption (Assumption 1.1), the Equidistant Surface Transversality Assumption (Assumption 2.12), and the Extended Boundedness Assumption (Assumption 3.30), the boundary of a linked second order generalized Voronoi region is connected.*

Proof: The boundary of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ comprises

- one generalized Voronoi edge (Assumption 3.30, Lemma 3.31),
- zero or more GVG^2 equidistant edges,
- zero or more boundary edge fragments from the same boundary edge,
- zero or more floating boundary edge fragments from the same floating boundary edge, and
- zero or more occluding edge fragments from the same occluding edge.

A boundary of a second order generalized Voronoi region can be written as

$$\partial \mathcal{F}_k\Big|_{\mathcal{F}_{ij}} = \mathcal{F}_{ijk} \bigcup_{l_1 \in L_1} \mathcal{F}_{kl_1}\Big|_{\mathcal{F}_{ij}} \bigcup_{l_2 \in L_2} c_{ij}^{l_2} \bigcup_{l_3 \in L_3} f c_{ij}^{l_3} \bigcup_{l_4 \in L_4} \bigcup_{l_5 \in L_5} v_{il_4}^{l_5},$$

where L_i are the sets of indices that catalog the appropriate structure types that are contained in the boundary of the generalized Voronoi region, $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$.

Since $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ is a closed and connected set (treat each connected component separately), the boundary of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ can be expressed as the union of connected components:

$$\partial \mathcal{F}_k\Big|_{\mathcal{F}_{ij}} = \bigcup_i \partial_i \mathcal{F}_k\Big|_{\mathcal{F}_{ij}},$$

where $\partial_i \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ is the ith connected boundary component.

By Lemmas 3.18, 3.19, and 3.31, $\mathcal{F}_{ijk}$ can only be the subset of one of the boundary components. Let $\partial_1 \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ be the connected component that contains $\mathcal{F}_{ijk}$ (i.e., $\mathcal{F}_{ijk} \subset \partial_1 \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$). Furthermore, by the Extended Boundedness Assumption (Assumption 3.30) and Proposition 3.17, $\mathcal{F}_{ijk} \neq \partial_1 \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$, because $\mathcal{F}_{ijk}$ cannot be a cycle. Finally, since $\mathcal{F}_{ijk}$ is not a cycle, Lemma 3.8 guarantees that second order generalized Voronoi edges will emanate from the meet points which are the end points of $\mathcal{F}_{ijk}$. Therefore, $\partial_1 \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ contains a GVG edge and GVG^2 equidistant edge, and perhaps boundary, floating boundary, and occluding fragments. That is,

$$\partial_1 \mathcal{F}_k\Big|_{\mathcal{F}_{ij}} = \mathcal{F}_{ijk} \bigcup_{l_1 \in L_1} \mathcal{F}_{kl_1}\Big|_{\mathcal{F}_{ij}} \bigcup c_{ij} \bigcup f c_{ij} \bigcup v_{ij},$$

where $L_1 \subset L$ and $\bigcup_i L_i = L$.

For $i > 1$, the boundary components can have one of the following forms:

$$\partial_i \mathcal{F}_k\Big|_{\mathcal{F}_{ij}} = \begin{cases} \bigcup_{l_i \in L_i} \mathcal{F}_{kl_i}\Big|_{\mathcal{F}_{ij}} \bigcup c_{ij} \bigcup f c_{ij} \bigcup v_{ij} & \text{(GVG}^2\text{ period with GVG}^2\text{ edges)}, \\ V_{ij} & \text{(occluding period)}, \\ C_{ij} & \text{(boundary period)}. \end{cases}$$

The Extended Boundedness Assumption (Assumption 2.12) and Lemma 3.33 assert that for $i > 1$, $\partial_i \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ may only be an "inner" GVG^2 period. If no such structure exists, then $\partial_1 \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ is the only connected component of the second order generalized Vorononoi region, and the proof is done.

However, if there is an inner GVG^2 period, Lemmas 3.37, 3.39, and 3.41 guarantees a link can be constructed between the inner GVG^2 period to $\partial_1 \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. Therefore, $\partial_1 \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$ is either the only connected boundary component of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$, or there exists other connected boundary components which readily link to $\partial_1 \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$, and thus the boundary of the linked second order generalized Voronoi region is connected. ■

3.4.4 The L^{3}HGVG is Connected

Proposition 3.43 *Given the Boundedness Assumption (Assumption 1.1), Equidistant Surface Assumption (Assumption 2.12) and Extended Boundedness Assumptions (Assumption 3.30), the L^{3}HGVG is connected.*

Proof: The proof of L^{3}HGVG's connectivity is done in two steps. The first step shows that each "linked" second order generalized Voronoi region is a generalized Voronoi cell, and the second step shows that the union of the generalized Voronoi regions form a generalized Voronoi complex on each two-equidistant face. Since the union of the two-equidistant faces is connected (Section C.4), the union of the roadmaps on each of the two-equidistant faces is connected.

The previous sections showed that the second order generalized Voronoi regions are generalized Voronoi cells (Proposition 3.42). Therefore, the second order generalized Voronoi regions on a two-equidistant face form a generalized Voronoi complex because the following are immediately true:

1. $\bigcup_k \mathcal{F}_k\Big|_{\mathcal{F}_{ij}} = \mathcal{F}_{ij}$,

2. $\text{int}(\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}) \bigcap \text{int}(\mathcal{F}_l\Big|_{\mathcal{F}_{ij}}) = \emptyset \quad \forall k, l$,

3. $\mathrm{cl}(\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}) \bigcap \mathrm{cl}(\mathcal{F}_l\Big|_{\mathcal{F}_{ij}}) \neq \emptyset \iff \dot{\partial}\mathcal{F}_k\Big|_{\mathcal{F}_{ij}} \bigcap \dot{\partial}\mathcal{F}_l\Big|_{\mathcal{F}_{ij}} \neq \emptyset.$

■

3.4.5 Proofs of Lemmas in Section 3.4.2

Proof of Lemma 3.37

Lemma 3.37. *If an inner GVG^2 period with GVG^2 edges exists on the boundary of the second order generalized Voronoi region, then a link exists from the outer boundary to it.*

Proof: By the Extended Boundedness Assumption (Assumption 3.30), if an inner GVG^2 period contains a GVG^2 edge, then it must contain a second order meet point, $\mathcal{F}_{kl_1l_2}\Big|_{\mathcal{F}_{ij}}$, such that (1) at this point, obstacles C_i and C_j are the closest obstacles and obstacles C_k, C_{l_1}, and C_{l_2} are the second closest, and (2) there exists a point, x, on the outer boundary where $d_{l_1}(x) = d_{l_2}(x)$. A link can be formed by tracing $\mathcal{SS}_{l_1l_2}\Big|_{\mathcal{F}_{ij}}$, a second order two-equidistant surjective surface constrained to a two-equidistant face, past a second order meet point, $\mathcal{F}_{kl_1l_2}\Big|_{\mathcal{F}_{ij}}$. By the Boundedness Assumption (Assumption 1.1) and Lemma 3.15, $\mathcal{SS}_{l_1l_2}\Big|_{\mathcal{F}_{ij}}$ is guaranteed to be unbounded and therefore must intersect another boundary component of $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$. ■

Proof of Lemma 3.39

Lemma 3.39. *In $\mathbb{R}^3$, if a boundary period exists on the boundary of the second order generalized Voronoi region, then it must be an "inner boundary" and a link exists from the outer boundary to it.*

Proof: This proof relies on the following lemma.

Lemma 3.27 *If the objects $C_{i_1} \ldots, C_{i_k}$ intersect, then the associated k-equidistant surjective surface, $\mathcal{SS}_{i_1 \ldots i_k}$, is unbounded. In fact, if objects $C_{i_1}, \ldots, C_{i_k}$ intersect,* *then none of the gradients,* $\nabla d_{i_1}(x), \ldots, \nabla d_{i_k}(x)$, *is orthogonal to* $T_x \mathcal{SS}_{i_1 \ldots i_k}$ *for all* $x \in \mathcal{SS}_{i_1 \ldots i_k}$.

Lemma 3.27 guarantees that a two-equidistant surjective surface, $\mathcal{SS}_{ij}$, is unbounded. By hypothesis, it has only one boundary, C_{ij}. When C_{ij} is a Jordan Curve, the bounded portion of $\mathcal{SS}_{ij}$ is the empty set, and the unbounded portion of $\mathcal{SS}_{ij}$ is $\mathcal{SS}_{ij}$. Therefore, the generalized Voronoi region, $\mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$, which has C_{ij} on its boundary, lies in the unbounded portion of $\mathcal{SS}_{ij}$. Therefore, C_{ij} is an inner boundary.

At a point $x \in C_{ij}$, $d_i(x) = d_j(x) = D(x) = 0$. Let $Y = (\text{nbhd}(x) \bigcap \mathcal{F}_{ij}) \backslash C_{ij}$ for a sufficiently small nbhd(x). Therefore, a path from x to any point $y \in Y$ increases the distance to objects C_i and C_j. That is, following $\nabla d_i(x)$ or $\nabla d_j(x)$ brings the robot to a point $y \in Y$. Let the second closest obstacle be C_k (i.e., $y \in \text{int}(\mathcal{F}_k\Big|_{\mathcal{F}_{ij}})$).

Now a path from y must be traced. By Lemma 3.27, the generalized gradient of D never vanishes on $\mathcal{SS}_{ij}$ (and thus $\mathcal{F}_{ij}$). In fact, using nonsmooth analysis, Proposition 2.21 states that $\pi_{\mathcal{F}_{ij}} \partial D(x) = \pi_{\mathcal{F}_{ij}} \nabla d_i(x)$ for all $x \in \mathcal{F}_{ij}$. Therefore, gradient ascent using the gradient of D on $\mathcal{F}_{ij}$, normally a nonsmooth function, is reduced to gradient ascent using $\nabla d_i(x)$ projected onto $\mathcal{F}_{ij}$. The gradient $\nabla d_i(x)$, projected onto $\mathcal{F}_{ij}$ never vanishes.

Since distance is increasing, a path traced out by following the gradient $\nabla d_i(x)$ never encounters the boundary period; in fact, such a path will be maximally moving away from the boundary period. By the boundedness assumption (Assumption 1.1), the path terminates on one of the following: (1) at a point x where there exists an object C_l for which $d_l(x) = d_k(x) \geq d_i(x) = d_j(x)$, or (2) at a point x where $d_i(x) = d_j(x) = d_k(x)$. ■

Proof of Lemma 3.41

Lemma 3.41. *In* $\mathbb{R}^3$, *if an occluding period exists on the boundary of a second order generalized Voronoi region, then a link can be made to the period.*

Proof: This proof relies on the following useful property of two-equidistant surjective surfaces.

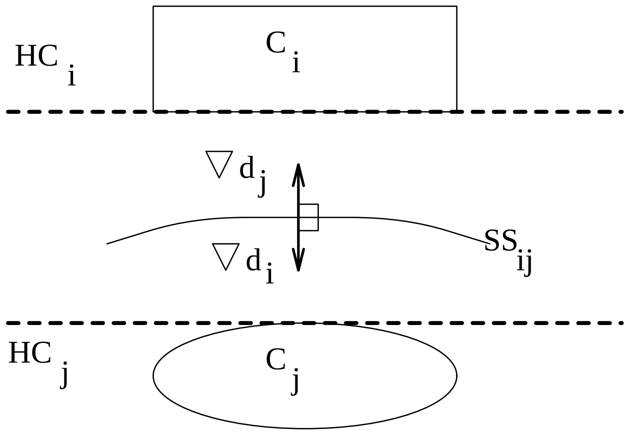

Figure 3.37 The supporting planes for convex obstacles C_i and C_j.

Lemma 3.44 *Except at a local minima of the multi-object distance function, the generalized gradient of the multi-object distance function never projects to zero on any tangent space of a two-equidistant surjective surface.*

Proof: Let $\pi_{T_x \mathbb{SS}_{ij}}$ be the orthoginal projection operator onto $T_x \mathbb{SS}_{ij}$. Let l be the shortest distance between two convex obstacles C_i and C_j. Therefore, for all $x \in \mathbb{SS}_{ij}$, $D(x) \geq l$. Assume that there exists a point x where $\pi_{T_x \mathbb{SS}_{ij}} \partial D(x) = 0$. By hypothesis, $D(x) = L > l$. By Proposition 2.21, $\pi_{T_x \mathbb{SS}_{ij}} \nabla d_i(x) = \pi_{T_x \mathbb{SS}_{ij}} \nabla d_j(x) = 0$. That is, $\nabla d_i(x)$ and $\nabla d_j(x)$ are each orthogonal to $T_x \mathbb{SS}_{ij}$. In fact, by definition of the two-surjective equidistant surface, $\nabla d_i(x) = -\nabla d_j(x)$.

Let $HC_i(x)$ and $HC_j(x)$ be two supporting hyperplanes of C_i and C_j respectively such that they are orthogonal to $\nabla d_i(x)$ and $\nabla d_j(x)$, respectively, and pass through the nearest points in C_i and C_j to x, respectively. See Figure 3.37.

Since $\nabla d_i(x) = -\nabla d_j(x)$, $HC_j(x)$ is orthogonal to $\nabla d_i(x)$ as well. Therefore, $HC_i(x)$ and $HC_j(x)$ are parallel. Thus, the distance between convex obstacles C_i and C_j can never be less than L. This is a contradiction.

▼

Let x be a point in the occluding fragment $v_{kl}\Big|_{\mathcal{F}_{ij}}$. Without loss of generality, assume that $x \in \text{int}(v_{kl}\Big|_{\mathcal{F}_{ij}})$. Recall that the X-distance function measures distance to obstacles, even if they are occluded. Assume that $d_l^X(x) > d_k^X(x) > d_i^X(x) = d_j^X(x)$. For a neighborhood $Y = \text{nbhd}(x) \bigcap \mathcal{F}_{ij}$, there are two connected subsets: Y^l, points $y \in Y$ for which $d_l^V(y) > d_k^V(y) > d_i^V(y) = d_j^V(y)$, and Y^k, points $y \in Y$ for which $d_k^V(y) > d_l^V(y) > d_i^V(y) = d_j^V(y)$. For points where $d_k^V(y) > d_l^V(y)$, $d_k^V(y)$ is infinite. This occurs when C_i or C_j occlude C_k. Without loss of generality assume that C_i occludes C_k.

Since we are using convex obstacles, without loss of generality, we can let x_{min} be the unique isolated local minima of D on $\mathcal{SS}_{ij}$. That is, for all $x \in \mathcal{SS}_{ij} \backslash \{x_{min}\}$, $D(x) > D(x_{min})$. Furthermore, $d_i(x) > d_i(x_{min})$ for all $x \in \mathcal{SS}_{ij} \backslash \{x_{min}\}$.

Recall that $\nabla d_i(x_{min})$ is orthogonal to $T_{x_{min}} \mathcal{SS}_{ij}$. In other words, there exists a straight line, L, between x_{min} and the nearest point on C_i to x_{min}, such that L is orthogonal to $T_{x_{min}}$, C_i, and C_j.

Without loss of generality, assume that $x_{min} \notin v_{kl}\Big|_{\mathcal{F}_{ij}}$. So, x_{min} is either inside or outside the occluding period. However, since there exists a straight line from x_{min} to C_i, C_i must occlude C_k at x_{min}. Therefore, x_{min} is inside the occluding period.

Lemma 3.44 guarantees that the gradient of D does not vanish on a two-equidistant face (except at the local minima) and Proposition 2.21 states that the generalized gradient of D, constrained to a two-equidistant face, is a vector, not a set. Therefore, if the occluding period is an inner boundary component of a second order generalized Voronoi region, a link from an occluding period to an outer boundary component can be made by following the gradient of the multi-object distance function, constrained to the two-equidistant face on which the occluding period is defined. Generating a link to an inner occluding period is accomplished by following the negated gradient of D.

When the occluding period is an outer boundary component, a link from (to) an occluding boundary is made by following the negated gradient (gradient) of the multi-object distance function, D. ■

Chapter 4

Incremental Construction of the GVG

A key feature of the L^3HGVG is that it can be incrementally constructed using environmental features that are within line of sight of the robot. In the scenario in which the robot has no a priori information about the environment, the robot must construct a roadmap in an incremental manner because most environments do not contain one vantage point from which a robot can "see" the entire world, and thereby construct a roadmap from a single vantage point. The incremental construction techniques, described in this chapter, provide a rigorous approach to constructing the L^3HGVG using only line of sight sensory information. Although the proof of connectivity of the L^3HGVG is limited to the three-dimensional case, the incremental construction techniques described in this chapter are general.

Although there do exist other sensor based schemes (see [35] for a review of planar sensor based planning), very little analysis of the numerical properties of these schemes has been done. However, the incremental construction techniques, described in this section, are numerically well posed, and these techniques can be adapted to other sensor based planning methods such as the Opportunistic Path Planner (described in [9, 38]).

Some sensor based planners do not possess a complete incremental construction strategy. For example, Canny and Lin's OPP [9] constructs part of its roadmap (the freeways) for a multi-dimensional workspace using local information, and is therefore partially incremental. However, the construction of "bridge curves," which guarantee the roadmap's connectivity, requires the identification of "interesting critical points." Complete prior knowledge of the world's geometry is needed to identify the critical points. This is a major limitation of their algorithm for sensor based implementation. Rimon and Canny [38] suggested a way to "sensorize" the OPP algorithm. They introduce the notion of a "critical point sensor," and a "minimum clearance" sensor, though the implementation of such sensors is not well detailed. Furthermore, they do not provide a detailed method to construct the freeway segments from sensor data.

In contrast, this chapter formulates a method for the construction of roadmap segments from sensor data. (It is worth noting that the incremental construction procedure can be the basis of a numerical method to construct a roadmap when full geometry of the world is available.) The HGVG's properties of accessibility, departability and connectivity translate to *incremental accessibility, incremental departability*, and *traceability,* respectively, in the incremental construction of the HGVG.

The incremental construction algorithms, described in this chapter, borrows techniques from the numerical curve tracing literature. Initially, the initial techniques were developed to generate GVG edges (Section 4.1), but then they were generalized to trace GVG^2 edges (Section 4.3) and to effect incremental accessibility (Section 4.2). The accessibility section may seem out of order because the traceability and accessibility sections use some common results which are easier to visualize in the traceability context. Next, the incremental linking procedure is described in Section 4.4 and the incremental departability procedure is described in Section 4.5. The entire algorithm is verified by simulations and experiments that are reviewed in Section 4.6, and Chapter 5, respectively.

4.1 Traceability of the GVG

In an incremental context, the property of connectivity is interpreted as *traceability.* More specifically, traceability implies that using only local data, the robot can "trace" the GVG (or HGVG) edges and determine when to terminate the tracing procedure. The robot concludes the edge tracing process when it encounters (1) a meet point, a point where GVG edges intersect (2) a boundary point, a point where a GVG edge intersects an obstacle, or (3) a floating boundary point, a point where two gradient vectors converge on each other. At each meet point, the robot begins tracing new edges. The robot also changes directions and traces boundary edges when it encounters a boundary point or a floating boundary point. In this section, we present and analyze a method for tracing a connected component of the GVG. For the sake of explanation, the following discussion is limited to the GVG and is generalized to the HGVG in the following section.

4.1.1 Properties for Tracing

Naively, one could trace an edge by repeated application of the accessibility method. That is, the robot would move a small distance along a given direction—either a fixed direction, or perhaps the tangent direction to the current edge. Gradient ascent would then be used to move back onto the local edge. The OPP [9] method and its sensor based adaptation [38] use this strategy with a fixed stepping direction. However, gradient ascent can be a computationally expensive procedure because of its slow convergence. Also, the constant step direction leads to undesirable roadmap artifacts [15].

Our approach borrows some basic ideas and techniques from numerical continuation methods [24]. Continuation methods are used to trace the roots of the expression $G_1(y, \lambda) = 0$ as the parameter λ is varied.

The incremental construction of a GVG edge can be implemented as follows. Let x be a point on the GVG. Choose local coordinates at x so that the first coordinate, z_1, lies in the direction of the tangent to the graph at x (see Figure 4.1). At x, let the hyperplane spanned by coordinates $z_2, \ldots, z_m$ be termed the "normal plane." We can thus decompose the local coordinates into $z = (y, \lambda)$, where $\lambda = z_1$ is termed the "sweep" coordinate and $y = (z_2, \ldots, z_m)$ are the "slice" coordinates. Now define

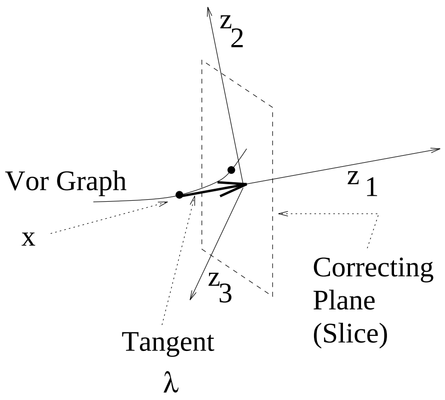

Figure 4.1 Continuation Method.

the function $G_1 \colon \mathbb{R}^{m-1} \times \mathbb{R} \to \mathbb{R}^{m-1}$ as follows:

$$G_1(y, \lambda) = \begin{bmatrix} (d_1 - d_2)(y, \lambda) \\ (d_1 - d_3)(y, \lambda) \\ \vdots \\ (d_1 - d_m)(y, \lambda) \end{bmatrix}. \tag{4.1}$$

The function $G_1(y, \lambda)$ assumes a zero value only on the GVG. Hence, if ∇G_1 is surjective at some point, then the implicit function theorem implies that the roots of $G_1(y, \lambda)$ locally define a generalized Voronoi edge as λ is varied. By numerically tracing the roots of this function, we can locally construct an edge.

While there are a number of curve tracing techniques [24], we use an adaptation of a common predictor-corrector scheme, as illustrated in Figure 4.1. Assume that the robot is located at a point x on the GVG. The robot takes a "small" step, $\Delta\lambda$, in the z_1-direction (i.e., the tangent to the local GVG edge). In general, this *prediction step* takes the robot off the GVG. Next, a *correction method* is used to bring the robot back onto the GVG. If $\Delta\lambda$ is small, then the graph will intersect a *correcting plane,* which is a plane orthogonal to the z_1-direction at a distance $\Delta\lambda$ away from the origin. The correction step finds the location where the GVG intersects the correcting plane.

Let $\nabla_y G_1$ be the matrix formed by taking the derivative of Equation 4.1 with respect to the normal plane's coordinates (slice coordinates). It takes the form

$$\nabla_y G_1(y, \lambda) = \begin{bmatrix} (\nabla_y d_1(y,\lambda) - \nabla_y d_2(y,\lambda))^T \\ (\nabla_y d_1(y,\lambda) - \nabla_y d_3(y,\lambda))^T \\ \vdots \\ (\nabla_y d_1(y,\lambda) - \nabla_y d_m(y,\lambda))^T \end{bmatrix}, \tag{4.2}$$

where ∇_y denotes the gradient with respect to the y-coordinates. We will show that $\nabla_y G_1(y, \lambda)$ is full rank at every (y, λ) in a small enough neighborhood of the GVG, and so it is possible to use an iterative Newton's Method to implement the corrector step. If y^h and λ^h are the hth estimates of y and λ, the $h+1$st iteration is defined as

$$y^{h+1} = y^h - (\nabla_y G_1)^{-1} G_1(y^h, \lambda^h), \tag{4.3}$$

where $\nabla_y G_1$ is evaluated at (y^h, λ^h). After taking the prediction step, the goal of the correction step is to find where the GVG locally intersects the "correcting plane."

There are several things worth noting about this method. First, to evaluate $G_1(y, \lambda)$ and $\nabla_y G_1(y, \lambda)$, one only needs to know the distance and direction to the m objects that are closest to the robot's current location—information that is easily obtained from local distance sensor data. Second, Newton methods are quadratic in their convergence, and thus they would be substantially faster than the naive gradient ascent techniques. Third, $\nabla_y G_1(y, \lambda)$ is an $(m-1) \times (m-1)$ matrix, and is thus typically quite small in size (e.g., a scalar for two-dimensional environments, or a 2×2 matrix for three-dimensional environments), and the method is not computationally burdensome.

The following two subsections demonstrate that Equation 4.3 is well posed because $(\nabla_y G_1(y, \lambda))^{-1}$ is defined, and that we can always compute (using local sensor data) a vector which is tangent to the GVG. In proving these assertions, several new and useful properties of the generalized Voronoi graph are presented.

Computing the Tangent to the Graph.

We first tackle the question of how to determine the tangent to a GVG edge from sensor data. Recall that the *regular Voronoi graph (RVG)* is the Voronoi graph for the case in which the obstacles are points. Further, recall that the *regular k-equidistant face*, $\mathcal{R}_{i_1 \ldots i_k}$, is a k-equidistant face whose k closest obstacles are points. In m dimensions, a regular m-equidistant face is an RVG edge and it is equidistant to m closest point objects. Previously in the literature, such a structure was termed a Voronoi graph edge, but here, we will label it a *regular* Voronoi edge to better distinguish it from a generalized Voronoi edge.

Let x be a point on the GVG edge, and let $\{c_i\}$ denote the set of closest points of the m closest obstacles, $\{C_i\}$, to x. The RVG edge defined by the points $\{c_i\}$ and the GVG edge coincide at x. We can compute many items of interest about the GVG by exploiting the coincidence of the RVG with the GVG at x.

Proposition 4.1 *The tangent to a GVG edge at x is defined by the vector orthogonal to the hyperplane which contains the m closest points, $c_1, \ldots, c_m$, of the m closest objects, $C_1, \ldots, C_m$.*

Proof: This proposition is a simple consequence of the following two lemmas when $k = m$.

Lemma 2.19 *Let $c_1, \ldots, c_k$ be the k closest obstacle points to a point x. Let $\mathcal{R}_{i_1 \ldots i_k}$ be the regular Voronoi graph edge defined by these points where $k \leq m$. When Assumption 2.12 is upheld, any vector in the tangent space $T_x\mathcal{R}_{i_1 \ldots i_k}$ is orthogonal to the $k-1$-dimensional plane containing $c_1, \ldots, c_k$. The tangent space $T_x\mathcal{R}_{i_1 \ldots i_k}$ is also orthogonal to the $(k-1)$-dimensional plane containing the heads of the gradient vectors based at x.*

Lemma 2.20 *Let $c_1, \ldots, c_k$ be the closest points in the k nearest obstacles to $x \in \mathcal{F}_{i_1 \ldots i_k}$. At a point x in the k-equidistant face, the tangent space $T_x\mathcal{F}_{i_1 \ldots i_k}$ is the same as the tangent space $T_x\mathcal{R}_{i_1 \ldots i_k}$, where $\mathcal{R}_{i_1 \ldots i_k}$ is the regular k-equidistant face defined by $c_1, \ldots, c_k$.*

Let x be a point on a GVG edge defined by the obstacles $C_1, \ldots, C_m$. The

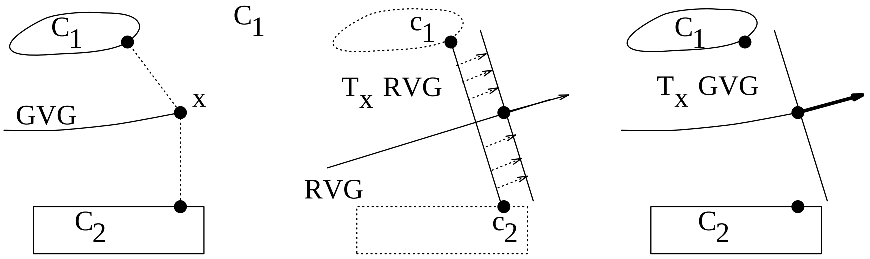

Figure 4.2 The tangent space is the line orthogonal to the line that connects the two closest points on the two closest obstacles.

m closest points $c_1, \ldots, c_m$ of the m closest obstacles define an RVG edge. When $k = m$, Lemma 2.19 asserts that the tangent space of the RVG edge at x is a one-dimensional vector space whose basis vector is orthogonal to the hyperplane which contains the m closest points $c_1, \ldots, c_m$.

Lemma 2.20 contends that the tangent space at x of the RVG edge, defined by $c_1, \ldots, c_m$, is the same as the tangent space at x of the GVG edge defined by $C_1, \ldots, C_m$. Thus, by knowing the distance and direction to the m nearest points, the tangent to a generalized Voronoi graph edge is easily computed. ■

Example 4.2 (Tangent in Two-dimensional Workspace) Let C_1 and C_2 be the two closest obstacles to a point x on $\mathcal{F}_{12}$. Let c_1 and c_2 be the two closest points on the two closest obstacles. Pass a line through c_1 and c_2; parallel shift this line so it passes through x. The displaced line is the normal plane and the line orthogonal to the normal plane is the tangent space. See Figure 4.2. ♦

Example 4.3 (Tangent in Three-dimensional Workspace) Let C_1, C_2 and C_3 be the three closest obstacles to a point x on $\mathcal{F}_{123}$. Let c_1, c_2 and c_3 be the three closest points on the three closest obstacles. The tangent to the GVG at x is a vector which is normal to the plane defined by c_1, c_2 and c_3. Let $\overline{c}_{12}$ be the vector formed by subtracting c_2 from c_1. Let $\overline{c}_{13}$ be defined in a likewise manner. The normal to the plane which contains c_1, c_2, and c_3 is collinear with the vector $\overline{c}_{12} \times \overline{c}_{13}$. ♦

The Matrix $\nabla_y G_1$ is Invertible.

The following proposition illustrates that the numerical procedure defined by Equation 4.3 is well posed for $\Delta\lambda$ sufficiently small.

Proposition 4.4 (Equidistant Surface Full Rank Property) *$\nabla_y G_1(y, \lambda)$ has full rank (i.e., has rank $(m-1)$) on the correcting plane in a neighborhood of the GVG.*

Proof: The following two lemmas are necessary in showing $\nabla_y G_1(x)$ is full rank. These lemmas furnish a general result for the function $G\colon \mathbb{R}^m \to \mathbb{R}^q$ which is defined as

$$G(x) = \begin{bmatrix} (d_{i_1} - d_{j_1})(x) \\ \vdots \\ (d_{i_q} - d_{j_q})(x) \end{bmatrix}. \tag{4.4}$$

If for all $r_1, r_2 \in \{1, \ldots, n\}$,

$$\{i_{r_1}, j_{r_1}\} \neq \{i_{r_2}, j_{r_2}\}, \tag{4.5}$$

then $G^{-1}(0)$ represents the intersection of q *distinct* two-equidistant surjective surfaces, i.e., $G^{-1}(0) = \mathcal{SS}_{i_1 j_1} \bigcap \cdots \bigcap \mathcal{SS}_{i_q j_q}$. When the condition in Equation 4.5 is met and

$$\begin{aligned} q &= m - 1, & & \\ i_r &= 1, & &\text{for } r = 1, \ldots, m-1, \\ j_r &= r + 1, & &\text{for } r = 1, \ldots, m-1, \end{aligned} \tag{4.6}$$

$G^{-1}(0)$ is the intersection of $m-1$ two-equidistant surjective surfaces which gives rise to a GVG edge. In other words, $G(x) = G_1(x)$. From here, this proof is now a simple consequence of the following:

Lemma 4.5 *Consider the mapping $G\colon \mathbb{R}^m \to \mathbb{R}^q$ defined as*

$$G(x) = \begin{bmatrix} (d_{i_1} - d_{j_1})(x) \\ \vdots \\ (d_{i_q} - d_{j_q})(x) \end{bmatrix}. \tag{4.7}$$

The rank of $\nabla G(x)$ is q for all $x \in \mathcal{SS}_{i_1 j_1} \bigcap \cdots \bigcap \mathcal{SS}_{i_q j_q}$, when

$$\mathcal{SS}_{i_1 j_1} \bigcap \cdots \bigcap \mathcal{SS}_{i_q j_q} \neq \emptyset$$

and for all r_1, r_2, $\{i_{r_1}, j_{r_1}\} \neq \{i_{r_2}, j_{r_2}\}$. That is, each pair $\{i_r, j_r\}$ is unique.

Proof: First, consider the case when $q = 2$. In this case, the robot is either equidistant to three obstacles (e.g., $i_1 = 1, j_1 = 2, i_1 = 1$, and $j_2 = 3$) or two sets of two obstacles (e.g., $i_1 = 1, j_1 = 2, i_2 = 3$, and $j_2 = 4$). The respective tangent spaces of $\mathcal{SS}_{i_1 j_1}$ and $\mathcal{SS}_{i_2 j_2}$ are:

$$T_x \mathcal{SS}_{i_1 j_1} = \{v \in T_x \mathbb{R}^m : \langle \nabla(d_{i_1} - d_{j_1})(x), v\rangle = 0\},$$
$$T_x \mathcal{SS}_{i_2 j_2} = \{v \in T_x \mathbb{R}^m : \langle \nabla(d_{i_2} - d_{j_2})(x), v\rangle = 0\}.$$

By the Equidistant Surface Transversality Assumption from Section 2.3.2, we know that $\mathcal{SS}_{i_1 j_1} \pitchfork \mathcal{SS}_{i_2 j_2}$. Assume at some point x, $\nabla(d_{i_1} - d_{j_1})(x) = \kappa \nabla(d_{i_2} - d_{j_2})(x)$. By definition, for all $w \in T_x \mathcal{SS}_{ij}$, $\langle \nabla(d_{i_1} - d_{j_1})(x), w\rangle = 0$. Since $\nabla(d_{i_1} - d_{j_1})(x) = \kappa \nabla(d_{i_2} - d_{j_2})(x)$, for $w \in T_x \mathcal{SS}_{ij}$, $\langle \nabla(d_{i_2} - d_{j_2})(x), w\rangle = 0$. This implies that $T_x \mathcal{SS}_{i_1 j_1} = T_x \mathcal{SS}_{i_2 j_2}$ which violates the Equidistant Surface Transversality Assumption (Assumption 2.12). Therefore, $\nabla(d_{i_1} - d_{j_1})(x) \neq \kappa \nabla(d_{i_2} - d_{j_2})(x)$; that is they are linearly independent. It therefore follows that

$$\operatorname{rank} \begin{bmatrix} (\nabla(d_{i_1} - d_{j_1})(x))^T \\ (\nabla(d_{i_2} - d_{j_2})(x))^T \end{bmatrix} = 2.$$

Now, we consider the case where $q = 3$. Here, the robot may be equidistant to four obstacles, three sets of two obstacles, or three obstacles and an additional pair of obstacles. Consider the matrix,

$$\nabla G(x) = \begin{bmatrix} (\nabla(d_{i_1} - d_{j_1})(x))^T \\ (\nabla(d_{i_2} - d_{j_2})(x))^T \\ (\nabla(d_{i_3} - d_{j_3})(x))^T \end{bmatrix}.$$

The Equidistant Surface Transversality Assumption (Assumption 2.12) guarantees each row is pairwise linearly independent:

$$\begin{aligned} \nabla d_{i_1} - \nabla d_{j_1} &\neq \kappa_{12}\Big(\nabla d_{i_2} - \nabla d_{j_2}\Big), \\ \nabla d_{i_1} - \nabla d_{j_1} &\neq \kappa_{13}\Big(\nabla d_{i_3} - \nabla d_{j_3}\Big), \\ \nabla d_{i_2} - \nabla d_{j_2} &\neq \kappa_{23}\Big(\nabla d_{i_3} - \nabla d_{j_3}\Big). \end{aligned} \tag{4.8}$$

It remains to show that no one row is a linear combination of the other two. Again, we prove this by contradiction. Assume $\nabla(d_{i_1} - d_{j_1}) = \alpha(\nabla(d_{i_2} - d_{j_2})) + \beta(\nabla(d_{i_3} - d_{j_3}))$. By definition, for all $w \in T_x \mathbb{SS}_{i_1 j_1}$, $\langle \nabla(d_{i_1} - d_{j_1})(x), w\rangle = 0$. Thus,

$$\begin{aligned} &\nabla(d_{i_1} - d_{j_1}) = \alpha(\nabla(d_{i_2} - d_{j_2})) + \beta(\nabla(d_{i_3} - d_{j_3})), \\ &\Rightarrow \Big\langle \Big(\alpha\Big(\nabla(d_{i_2} - d_{j_2})\Big) + \beta\Big(\nabla(d_{i_3} - d_{j_3})\Big)\Big), w \Big\rangle = 0, \\ &\Rightarrow \Big\langle \Big(\Big(\nabla(d_{i_2} - d_{j_2})\Big) + \tfrac{\beta}{\alpha}\Big(\nabla(d_{i_3} - d_{j_3})\Big)\Big), w \Big\rangle = 0. \end{aligned}$$

Since by Equidistant Surface Transversality Assumption [16], for all $w \in T_x S_{i_1 j_1}$:

$$\begin{aligned} \Big\langle \Big(\nabla(d_{i_2} - d_{j_2})(x)\Big), w \Big\rangle &\neq 0 \\ \Big\langle \Big(\nabla(d_{i_3} - d_{j_3})(x)\Big), w \Big\rangle &\neq 0, \end{aligned}$$

we conclude that $\nabla(d_{i_2} - d_{j_2}) = \frac{\beta}{\alpha}\Big(\nabla(d_{i_3} - d_{j_3})\Big)$. However, this contradicts one of the three inequalities in Equation 4.8. Therefore, all the rows of ∇G are linearly independent of each other and $\text{rank}(\nabla G) = 3$.

The lemma follows by induction. Assume the matrix

$$\tilde{G}(x) = \begin{bmatrix} (\nabla(d_{i_1} - d_{j_1})(x))^T \\ \vdots \\ (\nabla(d_{i_{q-1}} - d_{j_{q-1}})(x))^T \end{bmatrix}$$

has a rank of $q-1$, and let $\mathcal{SS}_{i_q j_q}$ be a two-equidistant surjective surface defined by obstacles C_{i_q} and C_{j_q}. The remainder of the this proof follows by contradiction. Assume that $\nabla(d_{i_q} - d_{j_q}) = \sum_{r=1}^{q-1} \alpha_r(\nabla(d_{i_r} - d_{j_r}))$. At a point $x \in \mathcal{SS}_{i_q j_q}$, for all $w \in T_x \mathcal{SS}_{i_q j_q}$, w is orthogonal to $\nabla(d_{i_q} - d_{j_q})(x)$. Therefore,

$$\sum_{r=1}^{q-1} \langle \alpha_r(\nabla(d_{i_r} - d_{j_r})(x)), w \rangle = 0 \implies \sum_{r=1}^{q-1} \alpha_r(\nabla(d_{i_r} - d_{j_r})(x)) = 0. \tag{4.9}$$

It follows that

$$\nabla(d_{i_1} - d_{j_1})(x) = \sum_{r=2}^{q-1} \frac{\alpha_r}{\alpha_1}(\nabla(d_{i_r} - d_{j_r})(x)), \tag{4.10}$$

which leads to a contradiction because the rank of $\tilde{G}$ is $q-1$ (i.e., the rows of $\tilde{G}$ are linearly independent of each other). Therefore, $\nabla(d_{i_q} - d_{j_q}) \neq \sum_{r=1}^{q-1} \alpha_r(\nabla(d_{i_r} - d_{j_r}))$, and thus

$$\operatorname{rank}(G(x)) = \operatorname{rank} \begin{bmatrix} \nabla(d_{i_1} - \nabla d_{j_1})(x) \\ \vdots \\ \nabla(d_{i_q} - \nabla d_{j_q)(x)} \end{bmatrix} = q.$$

▼

Lemma 4.6 *Consider the mapping* $G \colon \mathbb{R}^m \to \mathbb{R}^q$ *defined as*

$$G(x) = \begin{bmatrix} (d_{i_1} - d_{j_1})(x) \\ \vdots \\ (d_{i_q} - d_{j_q})(x) \end{bmatrix}. \tag{4.11}$$

On the normal slice plane (and all planes parallel to it) rank$(\nabla_y G)$ = *rank* (∇G) *for* $x \in \mathcal{SS}_{i_1 j_1} \bigcap \cdots \bigcap \mathcal{SS}_{i_q j_q}$, *when* $\mathcal{SS}_{i_1 j_1} \bigcap \cdots \bigcap \mathcal{SS}_{i_q j_q} \neq \emptyset$ *and each pair* $\{i_r, j_r\}$ *is unique. That is, for all* r_1, r_2, $\{i_{r_1}, j_{r_1}\} \neq \{i_{r_2}, j_{r_2}\}$.

Proof: As a consequence of the pre-image theorem, each element (i.e., row) of G defines a two-equidistant surjective surface. Since $\mathcal{SS}_{i_1 j_1} \bigcap \cdots \bigcap \mathcal{SS}_{i_q j_q} \neq \emptyset$ and each pair $\{i_r, j_r\}$ is unique (i.e., for all r_1, r_2, $\{i_{r_1}, j_{r_1}\} \neq \{i_{r_2}, j_{r_2}\}$), no two components of G are the same. Therefore, when Assumption 2.12 is upheld, the pre-image theorem asserts that $G^{-1}(0)$ is a manifold with co-dimension q whose tangent space at a point $x \in \mathcal{SS}_{i_1 j_1} \bigcap \cdots \bigcap \mathcal{SS}_{i_q j_q}$ is the null space of $\nabla G(x)$ which is equal to $T_x(\mathcal{SS}_{i_1 j_1} \bigcap \cdots \bigcap \mathcal{SS}_{i_q j_q})$. Finally, let the *normal slice* be the q-dimensional plane orthogonal to the tangent space of $G^{-1}(0)$ at x.

Pick $r \in \{1, \ldots, q\}$. Let c_{i_r} and c_{j_r} be the two closest points on obstacles C_{i_r} and C_{j_r}, respectively, to x. By Lemmas 2.19 and 2.20, $T_x \mathcal{SS}_{i_r j_r}$ can be viewed as co-dimension one plane that is the locus of points equidistant to c_{i_r} and c_{j_r}.

Let $n_1, \ldots, n_{m-1}$ be an orthonormal basis for $T_x \mathcal{SS}_{i_r j_r}$ whose origin is the mid-point of the segment that connects c_{i_r} and c_{j_r}. In this coordinate frame,

$$\begin{aligned} x &= (x^1, \ldots, x^{m-1}, 0)^T \\ c_{i_r} &= (0, \ldots, 0, \alpha)^T \\ c_{j_r} &= (0, \ldots, 0, -\alpha)^T, \end{aligned}$$

where $\alpha = \frac{\|c_{i_r} - c_{j_r}\|}{2}$.

Let the *slice line*, sl_r, be the line which is orthogonal to $T_x \mathcal{SS}_{i_r j_r}$ and passes through x. That is,

$$sl_r \quad = x + \lambda v \quad \forall \lambda \in \mathbb{R}$$

where $v \in (T_x \mathcal{SS}_{ij})^{\perp}$. Let the *base line*, bl_r, be the line defined by c_{i_r} and c_{j_r}, i.e.,

$$\begin{aligned} bl_r &= \lambda(c_{j_r} - c_{i_r}) \qquad \forall \lambda \in \mathbb{R} \\ &= (0, \ldots, 0, \lambda)^T \quad \forall \lambda \in \mathbb{R}. \end{aligned}$$

By construction, bl_r is also orthogonal to $T_x\mathcal{SS}_{i_rj_r}$, i.e., for all $w \in T_x\mathcal{SS}_{ij}$,

$$\begin{aligned}\langle w, bl_r\rangle &= \langle (w^1, \ldots, w^{m-1}, 0)^T, (0, \ldots, 0, \lambda)^T\rangle \quad \forall\lambda \in \mathbb{R}\\ &= 0.\end{aligned}$$

Therefore, the slice line and the base line are parallel, and thus, for all $\lambda \in \mathbb{R}$, $bl_r = \lambda v$ where $v \in (T_x\mathcal{SS}_{ij})^{\perp}$.

Let π_{bl_r} be the orthogonal projection onto bl_r operator. By definition, $\pi_{bl_r}(c_{j_r} - c_{i_r}) = c_{j_r} - c_{i_r}$. From this, we can conclude that $(x - c_{i_r}) - (x - c_{j_r})$ is equal to the projection of itself onto bl_r. In other words, $(x - c_{i_r}) - (x - c_{j_r}) \in bl_r$, or

$$\begin{aligned}\pi_{bl_r}((x - c_{i_r}) - (x - c_{j_r})) &= \pi_{bl_r}(c_{j_r} - c_{i_r})\\ &= c_{j_r} - c_{i_r}\\ &= (x - c_{i_r}) - (x - c_{j_r}).\end{aligned} \tag{4.12}$$

Note that $(x - c_{i_r}) = -d_{i_r}(x)\nabla d_{i_r}(x)$ and $(x - c_{j_r}) = -d_{i_r}(x)\nabla d_{j_r}(x)$ (recall that $d_{i_r}(x) = d_{j_r}(x)$). Substitute these relationships into (4.12).

$$\begin{aligned}\pi_{bl_r}((x - c_{i_r}) - (x - c_{j_r})) &= (x - c_{i_r}) - (x - c_{j_r})\\ \pi_{bl_r}(-d_{i_r}(x)\nabla d_{i_r}(x) - (-d_{i_r}(x)\nabla d_{j_r}(x))) &= -d_{i_r}(x)\nabla d_{i_r}(x) - (-d_{i_r}(x)\nabla d_{j_r}(x))\\ d_{i_r}(x)\pi_{bl_r}(\nabla d_{i_r}(x) - \nabla d_{j_r}(x)) &= d_{i_r}(x)(\nabla d_{i_r}(x) - \nabla d_{j_r}(x))\\ \pi_{bl_r}(\nabla d_{i_r}(x) - \nabla d_{j_r}(x)) &= \nabla d_{i_r}(x) - \nabla d_{j_r}(x).\end{aligned}$$

Since the slice line is parallel to the base line,

$$\begin{aligned}\pi_{sl_r}(\nabla d_{i_r}(x) - \nabla d_{j_r}(x)) &= \pi_{bl_r}(\nabla d_{i_r}(x) - \nabla d_{j_r}(x))\\ &= \nabla d_{i_r}(x) - \nabla d_{j_r}(x).\end{aligned}$$

We can conclude that

$$\begin{bmatrix}\nabla(d_{i_1} - d_{j_1})(x)\\ \vdots\\ \nabla(d_{i_q} - d_{j_q})(x)\end{bmatrix} = \begin{bmatrix}\nabla_{sl_1}(d_{i_1} - d_{j_1})(x)\\ \vdots\\ \nabla_{sl_q}(d_{i_q} - d_{j_q})(x)\end{bmatrix},$$

and since slice plane Y is the span of $sl_1, \ldots, sl_q$,

$$\nabla G(x) = \nabla_Y G(x). \tag{4.13}$$

Therefore, $\text{rank}(\nabla G(x)) = \text{rank}(\nabla_Y G(x))$. ▼

The matrix ∇G_1 is an $m-1$ by $m-1$ matrix, and thus by Lemma 4.5, the rank of ∇G_1 is $m-1$. Lemma 4.6 asserts that $\text{rank}(\nabla_y G_1)$ is $m-1$ for all $x \in \mathfrak{F}^m$, and therefore must be invertible at x.

Since the rank operation is a continuous function, $\nabla_y G$ must be invertible in an open neighborhood around $x = (y, \lambda) \in \mathfrak{F}^m$. This open neighborhood will intersect the correcting plane for $\Delta\lambda$ sufficiently small, and thus $\nabla_y G$ is invertible on the correcting plane as well. ■

In practice, the neighborhood of invertability is quite large with this method. Practically speaking, this result states that the numerical procedure defined by Equation 4.3 will be robust for reasonable errors in robot position, sensor errors, and numerical round off errors.

4.1.2 Terminating Conditions

So far, we have shown that the robot can trace a generalized Voronoi edge, but when does a tracing procedure stop? Due to the boundedness of the robot's environment, the generalized Voronoi edges must terminate, as stated in the following proposition.

Proposition 4.7 *Given the Equidistant Surface Transversality Assumption, in a bounded environment, if a generalized Voronoi edge is not a cycle (a GVG edge diffeomorphic to a circle), it must terminate: (1) at a generalized Voronoi vertex (a meet point), (2) on the boundary of the environment, or (3) at a point where two gradients of single object distance functions become collinear.*

Proof: This proof is a consequence of Proposition 2.11 when $k = m$. ■

When the GVG edge is cycle, the edge tracing procedure terminates when the robot circumnavigates the cycle. This procedure requires that the robot possess an accurate dead reckoning system.

Incremental construction of the GVG is akin to a graph search method where the generalized Voronoi edges are the "edges" and the meet points and boundary points are the "nodes." Once the robot has accessed a point on the GVG, it begins tracing an edge. If the robot encounters a meet point, it marks the direction from where it came as explored, and then explores one of the other m edges that emanate from the meet point. It also marks that direction as being explored. If the robot encounters another unvisited meet point, the above procedure is recursively repeated. When the robot reaches a boundary point, it simply turns around and retraces its path to some previous meet point with unexplored directions. The robot terminates exploration of the GVG component (i.e., there may be other disconnected GVG component) when there are no more unexplored directions at any meet point. If the robot is looking for a particular destination whose coordinates is known, then the robot can invoke graph searching techniques such as the A-star algorithm, or depth first search algorithm.

Meet Point Detection

Finding the meet points is essential to proper construction of the graph. While a meet point occurs when the robot is equidistant to $m+1$ objects, it is unreasonable to expect that a robot can exactly detect such points. For example, while tracing an edge, it is unlikely that the robot will pass exactly through an $m+1$ equidistant point. Furthermore, sensor error may make such detection difficult. Nevertheless, as shown in Figure 4.3, meet points can be robustly detected by watching for an abrupt change in the direction of the (negated) gradients to the m closest obstacles. Such a change will occur in the vicinity of a meet point.

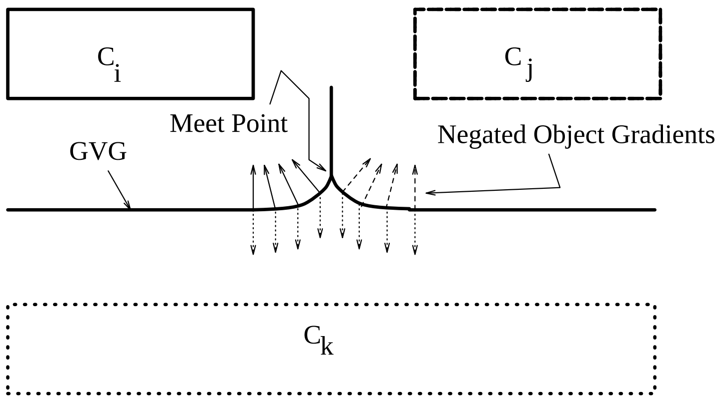

Figure 4.3 Meet Point Detection.

Departing a Meet Point

Recall that the robot is equidistant to $m+1$ objects at a meet point. It must be able to identify and explore the $m+1$ generalized Voronoi edges that emanate from each meet point in order to completely construct the GVG. Note that each emanating edge corresponds to an m-wise combination of the $m+1$ closest objects. Assume that we wish to explore and trace the edge corresponding to objects $C_1, \ldots, C_m$, the distances to which are $d_1(x) = d_2(x) = \cdots = d_m(x)$, respectively. Proposition 4.1 yields the one-dimensional tangent space to the generalized Voronoi edge corresponding to these m objects. If v is a basis vector of this GVG edge's tangent space, the robot must determine if it should depart the meet point in the $+v$ or $-v$ direction. We want the robot to move *away* from the $m+1^{st}$ obstacle, the distance to which is $d_{m+1}(x)$. If $\langle \nabla d_{m+1}, v\rangle > \langle \nabla d_i, v\rangle$ where $i \in \{1, \ldots, m\}$, then the robot should move in direction $+v$, otherwise $-v$. This effects motion away from C_{m+1}.

Now, the procedure for incremental construction of the GVG is complete. Rather than extending this description to the higher order GVG's, we consider the numerical aspects of incremental accessibility in the following section. Some of the results in the following section are useful for incrementally tracing all of the structures associated with the GVG^2.

4.2 Incremental Accessibility

Incremental Accessibility is the ability to access some point on the GVG via a collision-free path from any point in the free space, using only *local information.* It is obtained by gradient ascent of the multi-object distance function, D (Section 2.3.5). Recall from Section B.3 and [15], using nonsmooth analysis it can be shown that the *generalized gradient* of $D(x)$ is

$$\partial D(x) = \text{Co}\{\nabla d_i(x) \,:\, i \in I(x)\}, \tag{4.14}$$

where Co is the convex hull operation, and $I(x)$ is the set of indices such that $\forall i \in I(x)$, each C_i is the closest object to x (so there can be more than one "closest" object). Since $\partial D(x)$ is comprised of single object distance gradients, it can be readily *computed from sensor data.*

Naively, one can assume that gradient ascent of D reduces to a sequence of gradient ascent operations, constrained to equidistant faces where the robot travels via a collision-free path along a two-equidistant face, then a three-equidistant, and eventually to an m-equidistant face. That is, after the robot accesses a two-equidistant face, it performs gradient ascent of D, but constrained to the two-equidistant face until it reaches a three-equidistant face. Then, the robot performs gradient ascent of D but constrained to a three-equidistant face. This procedure is iteratively repeated until the robot encounters an m-equidistant face, a GVG edge.

Example 4.8 Figure 4.4 is a cross section of a three-dimensional world (imagine the polygons are coming out of the page) which contains two examples of accessibility in three dimensions. Starting from (A), the robot follows gradient ascent of d_j until it reaches $\mathcal{F}_{jk}$. From there, it does gradient ascent of $D = d_j = d_k$ constrained to $\mathcal{F}_{jk}$ until it reaches $\mathcal{F}_{ijk}$, an edge of the GVG. ♦

In actuality, the above "naive" accessibility procedure is not entirely correct in dimensions greater than three. The above procedure represents the tail end of a sequence of gradient ascent operations, each constrained to an equidistant face. It

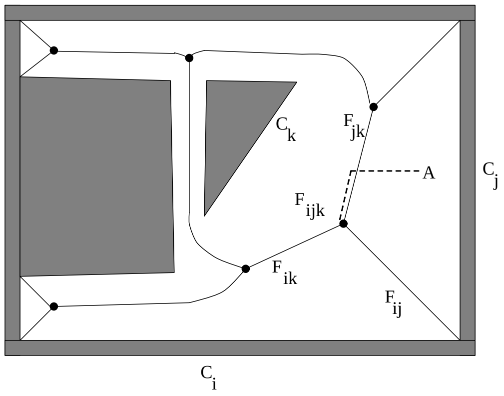

Figure 4.4 Gradient ascent accessibility in $\mathbb{R}^3$.

is possible that gradient ascent of D describes a path of the robot which traverses a two-equidistant face, then a three-equidistant face, then *another* two-equidistant face, then a three-equidistant face, etc. That is, in the course of doing gradient ascent of D, the robot may drop down to two in "equidistance count" before undergoing the cascading sequence of constrained gradient ascent operations which bring the robot to a GVG edge.

The procedure to trace a path on a k-equidistant face, using constrained gradient ascent of the multi-object distance function, borrows some basic ideas and techniques from numerical continuation methods [24], in a fashion similar to the approach described in Section 4.1. Here, the roots of the expression $G_A(y, \lambda) = 0$ as the "parameter" λ is varied describes a path on a k-equidistant face. Note, in this case λ is a vector, i.e., it is not a scalar as it was in the incremental traceability procedure.

Let x be a point on the k-equidistant face. Choose a local coordinate frame at x so that the first $m - k + 1$ coordinates, $(z_1, \ldots, z_{m-k+1})$, are the coordinates which span the tangent space of the k-equidistant face at x, and the next $k - 1$ coordinates, $(z_{m-k+2}, \ldots, z_m)$, span a plane termed the "normal slice plane." We can thus decompose the local coordinates into $z = (y, \lambda)$, where $\lambda = (z_1, \ldots, z_{m-k+1})$, the "sweep coordinates," and $y = (z_{m-k+2}, \ldots, z_m)$ are the "slice" coordinates. Note, there can be some confusion with this choice of coordinates; when $z = (y, \lambda)$, λ is the *first* $m-k+1$ coordinates and y is the *next* $k-1$ coordinates.

Now define the function $G_A\colon \mathbb{R}^{k-1}\times\mathbb{R}^{m-k+1}\to\mathbb{R}^{k-1}$ as follows:

$$G_A(y,\lambda)=\begin{bmatrix}(d_1-d_2)(y,\lambda)\\(d_1-d_3)(y,\lambda)\\\vdots\\(d_1-d_k)(y,\lambda)\end{bmatrix}. \tag{4.15}$$

The procedure for tracing a path on the k-equidistant face is similar to the approach described in Section 4.1. The robot starts at point on the k-equidistant face. At this point, and all others on the k-equidistant face, G_A vanishes. The robot takes a "small" step, $\Delta\lambda$, in the tangent space of the k-equidistant face such that $D(x)$ increases the most. Typically, this step takes the robot off of the k-equidistant face, and on a $(k-1)$-dimensional plane orthogonal to the k-equidistant face tangent space, the robot moves back onto the k-equidistant. This $(k-1)$-dimensional plane is called the "correcting plane." The correction step is the same as the one described in Section 4.1. If y^h and λ^h are the h^{th} estimates of y and λ, the $h+1^{st}$ iteration is defined as

$$y^{h+1}=y^h-\left(\nabla_y G_A\right)^{-1}G_A(y^h,\lambda^h) \tag{4.16}$$

where $\nabla_y G_A$ is evaluated at (y^h,λ^h). After taking the prediction step, the goal of the correction step is to find where the k-equidistant face locally intersects the correcting plane.

Again, it is important to note that in order to evaluate $G_A(y,\lambda)$ and $\nabla_y G_A(y,\lambda)$, one only needs to know the distance and direction to the k objects that are closest to the robot's current location—information that is easily obtained from local distance sensor data. The following propositions and lemmas demonstrate that this procedure is theoretically sound and can be implemented using local information.

Computing the Tangent Vector. The predictor step is a small step the direction in the tangent space of the k-equidistant face which maximally increases $D(x)$.

This step is determined in two steps: first the tangent space of the k-equidistant face at x is computed, and then the generalized gradient of D is projected onto it. Lemmas 2.19 and 2.20 furnish the tangent space and the following proposition shows how the generalized gradient is projected onto it. In fact, the following proposition states that the generalized gradient of D projects to a single vector on the tangent space.

Proposition 2.21 *The restriction of the multi-object distance function D to a k-equidistant face is smooth. That is, the generalized gradient of $D(x)$ projected onto $T_x\mathcal{F}_{i_1\ldots i_k}$ is equal to $\pi_{T_x\mathcal{F}_{i_1\ldots i_k}}\nabla d_i$ for all $i \in \{i_1 \ldots i_k\}$, where π is the orthogonal projection operator.*

Therefore, the robot takes the following step:

$$\begin{aligned}\pi_{\mathcal{F}_{i_1\ldots i_k}}\partial D(x) &= \pi_{\mathcal{F}_{i_1,\ldots,i_k}}\nabla d_{i_1}(x)\\ &= \nabla d_{i_1}(x) - \sum_{j=2}^{j=k}\frac{\nabla d_{i_1}(x)-\nabla d_{i_j}(x)}{\|\nabla d_{i_1}(x)-\nabla d_{i_j}(x)\|}\|\nabla d_{i_1}(x)\|.\end{aligned}$$

Computing the Correction Step. The correction procedure is guaranteed by

Proposition 4.9 *The matrix $\nabla_y G_A(y,\lambda)$ has full rank (i.e., has rank $(k-1)$) in a neighborhood of a k-equidistant face on the correcting plane.*

Proof: This is a simple consequence of Lemmas 4.5 and 4.6. Since $\nabla_y G_A$ is a $k-1$ by $k-1$ matrix, by these lemmas, it must have rank $(k-1)$ for $x \in \mathfrak{F}^k$, and therefore be invertible at x. Since the rank operation is a continuous function, $\nabla_y G_A$ must be invertible in an open neighborhood around $x = (y,\lambda) \in \mathfrak{F}^m$. This open neighborhood will intersect the correcting plane for $\|\Delta\lambda\|$ sufficiently small, and thus $\nabla_y G_A$ is invertible on the correcting plane as well. ■

4.3 Constructing the Second Order GVG

4.3.1 Second Order Generalized Voronoi Edges

The second (and higher) order GVG can be incrementally constructed in an analogous fashion. The key is to define a function whose roots define GVG^2 edges. The roots of the function

$$G_2(y,\lambda) = \begin{bmatrix} (d_1 - d_2)(y,\lambda) \\ (d_3 - d_4)(y,\lambda) \\ \vdots \\ (d_3 - d_m)(y,\lambda) \end{bmatrix} \tag{4.17}$$

locally trace out a GVG^2 equidistant edge. The first row of G_2 enforces equidistance between the closest objects C_1 and C_2. The remaining rows enforce equidistance between the second closest objects. Again, a predictor-corrector algorithm is used.

Computing the Tangent

The tangent to the GVG^2 edge is the null space of

$$\nabla_y G_2(y,\lambda) = \begin{bmatrix} (\nabla_y(d_1 - d_2)(y,\lambda))^T \\ (\nabla_y(d_3 - d_4)(y,\lambda))^T \\ \vdots \\ (\nabla_y(d_3 - d_m)(y,\lambda))^T \end{bmatrix}. \tag{4.18}$$

In $\mathbb{R}^3$, this can be easily computed with local sensor information. In $\mathbb{R}^3$,

$$\nabla_y G_2(y,\lambda) = \begin{bmatrix} (\nabla_y(d_1 - d_2)(y,\lambda))^T \\ (\nabla_y(d_3 - d_4)(y,\lambda))^T \end{bmatrix}. \tag{4.19}$$

In $\mathbb{R}^3$, the null space of $\nabla_y G_2(y, \lambda)$ is the set of vectors, v, for which

$$\langle \nabla_y (d_1 - d_2)(y, \lambda), v \rangle = 0 \quad \text{and} \quad \langle \nabla_y (d_1 - d_3)(y, \lambda), v \rangle = 0. \tag{4.20}$$

That is, the tangent to a GVG^2 equidistant edge is spanned by the vectors which are in the intersection of the tangent spaces to $\mathcal{SS}_{12}$ and $\mathcal{SS}_{13}$. The Equidistant Surface Transversality Assumption guarantees that these tangent spaces transversally intersect and thus their intersection is one-dimensional. A basis vector for this tangent space is $\nabla_y(d_1 - d_2)(y, \lambda) \times \nabla_y(d_1 - d_3)(y, \lambda)$. Since the tangent space is computed from the cross product of gradient vectors, the tangent space can be readily computed from sensor information.

The matrix ∇G_2 is invertible.

When

$$\begin{aligned} q &= m - 1, \\ i_1 &= 1, \\ j_1 &= 2, \\ i_r &= 3, && \text{for } r = 2, \ldots, m-1, \\ j_r &= r + 2, && \text{for } r = 2, \ldots, m-1, \end{aligned} \tag{4.21}$$

Lemmas 4.5 and 4.6 guarantee the matrix $\nabla_y G_2(y, \lambda)$ has full rank in a neighborhood of the GVG^2 on the correcting plane.

4.3.2 Boundary Edges

The incremental tracing of boundary edges requires that the robot moves along the perimeter of the environment where m obstacles meet in m dimensions. This can be done by tracing the roots of the following function, G_b, defined as

$$G_b(y, \lambda) = \begin{bmatrix} d_1\,(y,\lambda) - \epsilon \\ d_3\,(y,\lambda) - \epsilon \\ \vdots \\ d_{m-1}\,(y,\lambda) - \epsilon \end{bmatrix}, \tag{4.22}$$

where in this case ϵ is a small "safety" distance away from the environment. Continuity of the single object distance function guarantees there exists a small enough $\epsilon > 0$ such that the topology of the traced edges reflects that of the actual boundary edges. Later, we will see that G_b will be used in tracing edges of the *saturated generalized Voronoi graph,* which is a roadmap used when sensors function over a finite range, less than ϵ.

4.3.3 Floating Boundary Edges

Floating boundary edges are straight line segments and thus do not require complicated numerical curve tracing techniques. Proposition A.10 asserts that a floating boundary edge terminates at either a GVG edge end point or a boundary edge (C_{ij}) end point, which is a point where two convex obstacles (C_i and C_j) merge into one convex obstacles (e.g., such as the top of the box in Figure 3.4 in Example 3.3).

When a GVG edge terminates at a floating boundary edge point, the basis vector of the floating boundary edge must be determined. Let x^* be the point where the two gradient vectors converge and let v be the limiting vector of $\nabla d_i(x)$ (or $\nabla d_j(x)$) as x approaches x^*. The vector v is the basis vector of the floating boundary edge point, and after encountering the floating boundary edge point, the robot moves in the $-v$ direction. When a boundary edge terminates at a floating boundary edge, the robot moves in a direction v, as described above. (Note that when obstacles are polyhedra, then the floating boundary edge is a straight line extension and a boundary edge, which is also a straight line.)

4.3.4 Occluding Edges

The precise formulation of the incremental construction technique for occluding edges is still a topic of current research. The robot accomplishes occluding edge tracing by looking for discontinuities in range data while maintaining double equidistance between the two closest obstacles, C_i and C_j, when C_k and C_l be the second closest obstacles such that C_k occludes C_l on the occluding edge. As with the other procedures, the robot takes a prediction step which typically brings itself off of an occluding edge. Lemma 3.44 asserts that the robot corrects back onto the occluding edge by either performing gradient ascent or gradient descent of D: if C_k occlude C_l, the robot corrects using gradient ascent; otherwise the robot corrects using gradient descent. This approach only works when obstacles are smooth and the robot takes small steps in its prediction phase.

4.3.5 Terminating Conditions

In summary, the GVG^2 has the same terminating conditions as the GVG: a *second order meet point*, *second order boundary point*, *second order floating boundary point*, and a *second order cycle*. The second order meet points are detected in a fashion analogous to the (first order) meet points—the robot looks for a change in the gradients to the second nearest object, while maintaining equidistance to the two nearest objects. At a second order boundary point, the robot does not necessarily turn around and re-trace its steps to the previous second order meet point with unexplored directions. Instead, it traces both of the directions of the boundary edge it intersects (second order equidistant edges intersect boundary edges only in the interior).

4.4 Incremental Linking

Section 3.4.2 outlines three types of links: one involving an inner GVG^2 period, one involving a boundary period, and one involving a floating boundary period. Additionally, when the Extended Boundedness Assumption is relaxed, there is a need for a linking strategy to and from GVG cycles.

In Section 3.2.3, we introduced the notion of an inner and outer boundary for a second order generalized Voronoi region. There are situations in which the robot can infer if it is on an outer or inner boundary component. For example, while traversing a boundary component of a second order generalized Voronoi region, if the robot detects a boundary period, Lemma 3.39 implies that the robot is on an outer boundary component and the boundary period is an inner boundary component. However, in general, the robot may not know if it is on an outer or inner boundary component.

4.4.1 Inner GVG^2 Period Link

While traversing an inner boundary component, $\partial_I \mathcal{F}_k\Big|_{\mathcal{F}_{ij}}$, that contains GVG^2 equidistant edges and is disconnected from the outer boundary component, the robot builds a link outward from a meet point, $\mathcal{F}_{klp}\Big|_{\mathcal{F}_{ij}}$, which is formed by the edges $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$, $\mathcal{F}_{kp}\Big|_{\mathcal{F}_{ij}}$, and $\mathcal{F}_{pl}\Big|_{\mathcal{F}_{ij}}$. The link is the intersection of $\mathcal{SS}_{pl}\Big|_{\mathcal{F}_{ij}} \backslash \mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$. That is, instead of tracing the GVG^2 equidistant edge, the robot traces the points, x, where $d_l(x) = d_p(x) > d_k(x) > d_i(x) = d_j(x)$. Lemma 3.37 guarantees that this link brings the robot to an outer boundary component.

Unfortunately, the robot may not know initially if it is on an inner boundary component. Therefore, at all second order meet points, the robot must perform this procedure, which results in redundant links. The robot terminates the link tracing procedure when it encounters a GVG edge, a GVG^2 edge, a boundary edge, or a floating boundary edge. The terminating point becomes a node in the L^{3}HGVG.

Similarly, if the robot is on an outer boundary component, it must look for an additional pair of equidistant obstacles. However, the robot need not know if it is on an outer boundary component, so it must always perform the following linking strategy, once again resulting in redundant links. On a GVG edge, $\mathcal{F}_{ijk}$, the robot starts constructing a link when it encounters a point where two additional obstacles, C_l and C_p, are equidistant with the following distance relationship: $d_l(x) = d_p(x) = d_k(x) > d_i(x) = d_j(x)$. The robot terminates the link tracing process when it encounters a structure in the HGVG. If this structure is a second order meet point, the link is formed. Otherwise, the robot may backtrack to continue tracing the outer boundary component or save the link as another redundant structure.

A similar procedure is followed when a GVG^2 equidistant edge is on the outer boundary component. While tracing $\mathcal{F}_{kl}\Big|_{\mathcal{F}_{ij}}$, the robot's range sensor must look for equidistant between two obstacles, C_p and C_q. At this point, the robot traces a path where $d_p(x) = d_q(x) > d_k(x) > d_i(x) = d_j(x)$ on $\mathcal{F}_{ij}$ until it encounters an HGVG structure. Just like before, if this structure is a second order meet point, the link is formed. Otherwise, the robot may backtrack to continue tracing the outer boundary component or save the link as another redundant structure. A similar procedure exists for boundary edges and floating boundary edges.

4.4.2 Inner Boundary Edge Period

The linking procedure to an inner boundary period is a two-step process: detection of the inner period and then the explicit construction of the link. By Lemma 3.27, this linking procedure amounts to following a path defined by gradient descent of D on the second order generalized Voronoi region which contains the boundary period. Linking from the inner boundary period is accomplished via gradient ascent of D, constrained to a two-equidistant face (Section 4.4).

In order to describe the detection scheme, we define the *raw distance function*, which provides the distance to all the points on the boundary of the environment that are *within line of sight* of the robot. For the following definition, recall that S^{m-1} is an $(m-1)$-dimensional sphere embedded in $\mathbb{R}^m$. Sometimes we treat $s \in S^{m-1}$ as a point on an $(m-1)$-dimensional sphere, and other times we treat it as a unit vector whose head is in the $(m-1)$-dimensional sphere.

Definition 4.10 (Raw Distance Function) The distance between a point, $x \in \mathbb{R}^m$, and a point on an object that is within line of sight of x, in a direction $s \in S^{m-1}$. This is the length of the line segment $x + \lambda s$ and where $\lambda = \min_{\Lambda \in [0,\infty)} D(x + \Lambda s) = 0$.

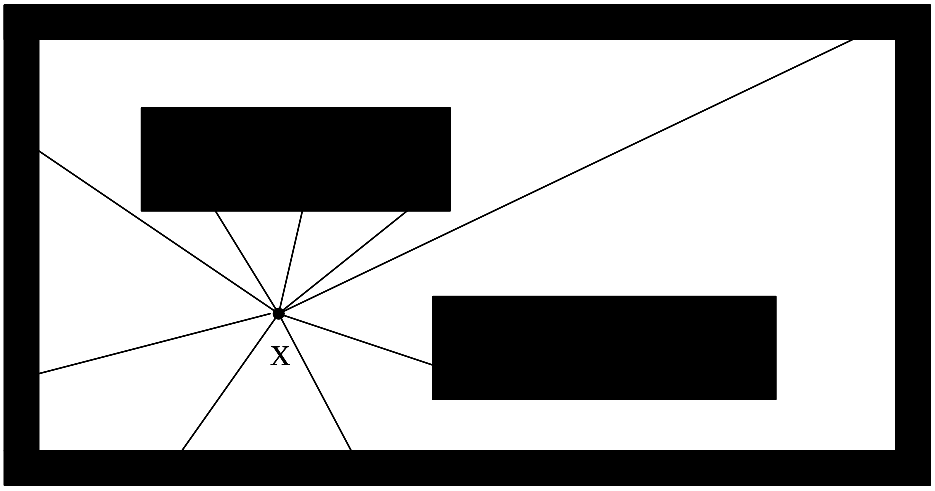

Figure 4.5 The solid lines are values of the raw distance function, $\rho(x, s)$, for a fixed $x \in \mathbb{R}^2$ and nine samples of $s \in S$. The filled regions are obstacles.

That is,

$$\rho(x, s) = \|x + \lambda s\| \quad \text{where } \lambda = \min_{\Lambda \in [0,\infty)} D(x + \Lambda s) = 0, \tag{4.23}$$

where D is the multi-object distance function which measures distance to the nearest point on the nearest obstacle.

A key feature of the raw distance function (Figure 4.5) is that it can be readily approximated by many realistic sensor configurations. The sensor measurement provides an approximate value of the distance function $\rho(x, s)$, and the direction to which the sensor is facing corresponds to the direction of measurement ($s \in S^{m-1}$). See Figure 2.2. We term this function the "raw distance function" because raw sensor readings approximate this function. The raw distance function is a necessary component for the experimental implementation of the GVG.

To detect a boundary edge from an outer boundary component, we look at the values of $\rho(x, s)$ restricted to the normal plane (a hyperplane orthogonal to the tangent vector at x on a GVG edge) with respect to s. It can be seen from Figure 4.6 that for convex polyhedra, if there exists a local maxima of $\rho(x, s)$ restricted to the normal plane, then there exists a point on a boundary edge. If for all points on

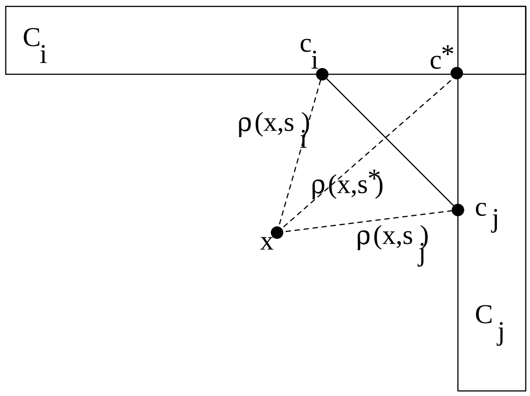

Figure 4.6 A local maxima of $\rho(x, s)$ with respect to s corresponds to a boundary point. The dotted lines delineate two values of the raw distance function on opposite sides of a local maxima. The solid line connects two points from two different convex sets, C_i and C_j.

the outer boundary component of a second order generalized Voronoi region, there exists a local maxima on each normal plane, then the outer boundary component surrounds an inner boundary period.

4.4.3 Occluding Period

The linking procedure to an occluding period is the same as it is for a boundary period. The detection scheme is similar; instead of looking for local maxima, the robot looks for discontinuities in the raw distance function restricted to a normal slice. If for all points on the outer boundary component of a second order generalized Voronoi region, there exists a discontinuity on each normal plane, then the outer boundary component surrounds an inner occluding period.

4.4.4 Links to Cycles

Linking away from outer GVG cycles is straightforward: simply apply one of the above techniques. Linking away from inner GVG cycles is also straightforward: go in any radially pointing outward direction. However, linking to inner GVG cycles is under current investigation. Such a linking procedure will be gradient descent of the distant of the second closest obstacle while maintaining double equidistantance

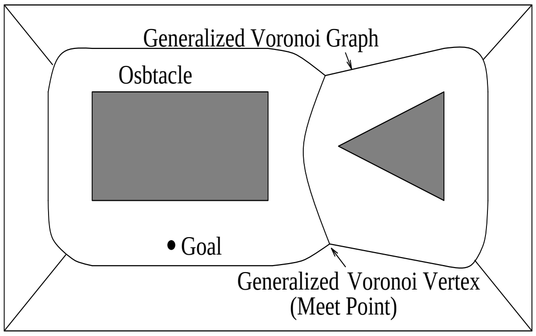

Figure 4.7 Original GVG.

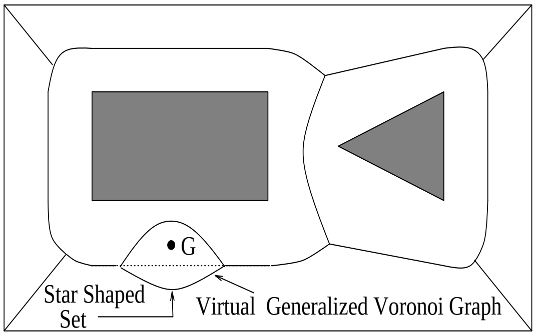

Figure 4.8 Virtual GVG.

to the closest obstacle (constrained gradient descent).

4.5 Incremental Departability

In sensor based exploration, the robot may or may not know the coordinates of its goal location. If the robot does not know the goal coordinates, it is assumed that the goal is defined by a beacon or other feature which the robot can detect once it is within line of sight of it. We therefore would like to find a departing method in which the robot can access the goal in a straight line. Treating the goal as an object, create a "virtual" Generalized Voronoi Graph (Figure 4.7). A star shaped set, bounded by the virtual GVG, surrounds the goal, and thus there exists a straight line path between any point on the boundary of this virtual star shaped set and the goal. Generally, the virtual GVG is connected to the GVG and thus there is a point within line of sight of the goal on the GVG. However, as we know from previous sections, the virtual GVG may be disconnected. In this case, it is necessary to build a link to the disconnected component that surrounds the goal. The linking strategy is a special case of the strategy one would use to link GVG cycles to other second order GVG edges.

4.6 Simulations

4.6.1 Planar Simulations

A planar simulator, the details and results of which are described in this section, validated this approach in the plane. The simulator reads in a list of polygons which serve as obstacles in an environment. The simulator also reads in a start point from which a point robot accesses the GVG. After accessing the GVG, the simulator traces a GVG edge using the numerical edge tracing techniques described in Section 4.1. The tracing procedure terminates when the robot detects a meet point or a boundary point, as described in Section 4.1.2. The simulator runs until all meet points are visited and their emanating edges have been explored.

The simulator maintains a list of meet points with unexplored directions. Each time the point robot encounters a new meet point, two new meet points (one for each new direction to explore) are added onto the front of the list. Each time the point robot encounters a boundary point, it terminates the edge tracing technique and starts tracing the edge associated with the meet point in the front of the list. This depth first searching procedure continues until the new meet point list is null.

The core of the planar simulator is the distance function which measures distance between a point and a convex polygon. The distance function used in the simulator is a modified version of Lin and Canny's closest feature algorithm described in [29]. Lin and Canny define a Voronoi region for each feature on the polygon. If a point is in the Voronoi region of a particular feature, then the distance to the polygon is the distance to the feature, which is either a point or a line segment.

The Voronoi region of an edge is the region in free space bounded by the edge, and the perpendiculars to the edge at the end points of the edge. The Voronoi region of a vertex is the region in free space which is bounded by the perpendiculars of the edges which meet at the vertex. See Figure 4.9 for an example of each region.

If the closest feature is a vertex, then the nearest point on the polygon is that vertex. If the closest feature is an edge, then the nearest point is in the interior of the edge. Let v_1 and v_2 be the end points of the edge and x be the point from which

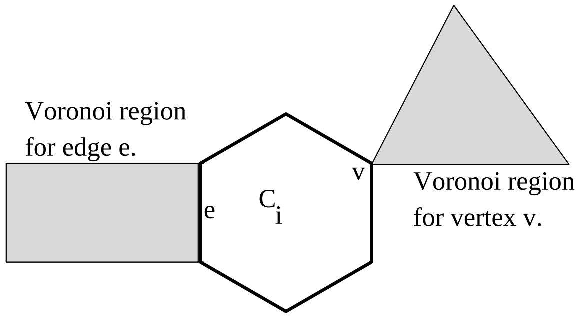

Figure 4.9 The shaded regions are Voronoi regions for their closest feature on obstacle C_i. The rectangular shaded region corresponds to the Voronoi region of the edge e; note, the Voronoi region is unbounded. The triangular shaded region corresponds to the Voronoi region of the vertex v, and the Voronoi region, here, is also unbounded.

distance is being measured. The nearest point is a point whose coordinates are the components of

$$v_1 + \langle x - v_1, \frac{v_2 - v_1}{\|v_2 - v_1\|}\rangle (v_2 - v_1).$$

The distance between a point x and a polygon is the two norm of x and the nearest point on the polygon. The single object distance gradient is the unit vector pointing away from the nearest point. Let c be the nearest point and thus the gradient is the vector $\frac{x-c}{\|x-c\|}$.

With the distance and gradient information in place, the simulator invokes the continuation methods described in Section 4.1. Figure 4.10 contains an environment populated with obstacles. The concave exterior is represented by the union of four rectangular polygons.

In Figure 4.11, the robot has accessed the GVG, traced one GVG edge, encountered a meet point, and continued tracing until a boundary point. The ticked solid lines represent the GVG in the plane (also the GVD); these are the locus of points equidistant to the two nearest obstacles. The ticks point to the nearest obstacles. Figures 4.12 and 4.13 display two more intermediate simulation results. Finally, Figure 4.14 shows the final simulation result.

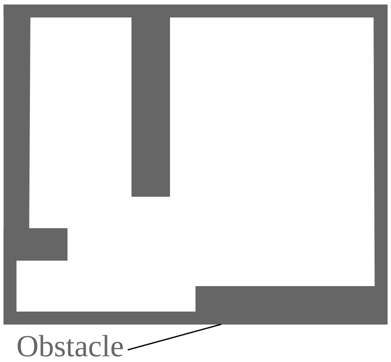

Figure 4.10 Floor plan of bounded environment.

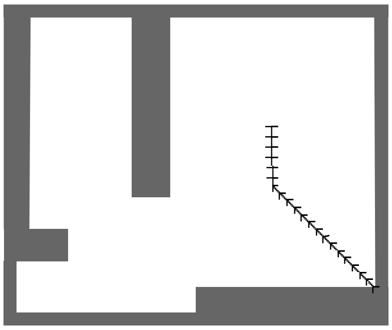

Figure 4.11 Iteration 1.

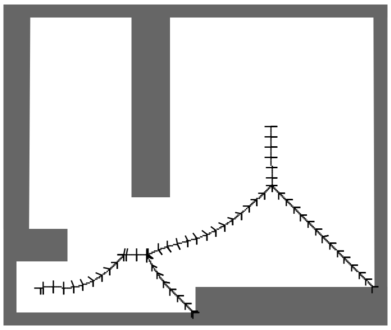

Figure 4.12 Iteration 5.

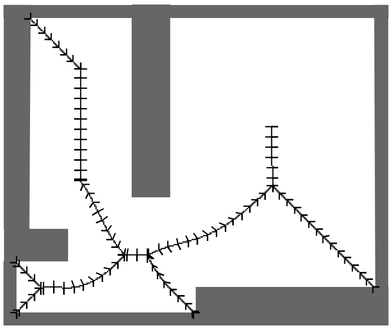

Figure 4.13 Iteration 10.

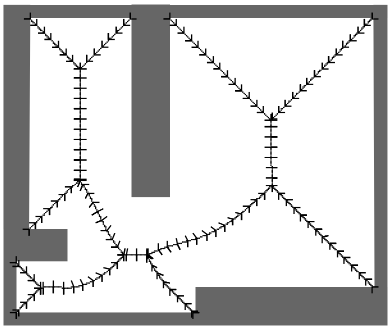

Figure 4.14 Iteration 14.

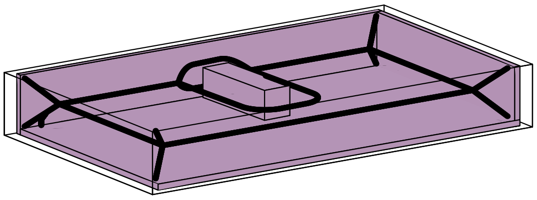

Figure 4.15 Results of applying the simulator to a three-dimensional box with a long box which is located off-center in the interior. Note that some of the walls were removed so the GVG lines, depicted as thick solid lines, can be displayed.

4.6.2 Three-Dimensional Simulator

A major advantage that the HGVG has over other methods is that it is applicable in higher dimensional workspaces. To this end, we have implemented a three-dimensional simulator which traces GVG edges. The algorithm and data structure of the three-dimensional simulator is similar to that of the planar version. The distance function code, used in three-dimensional simulator, was written by Brian Mirtich at Berkeley. Currently, the linking procedures (described in the next section) are under development. See Figure 4.15 for final results of GVG tracing.

4.7 Incremental Construction of the OPP

The incremental construction techniques described in this chapter can be readily extended to other roadmap algorithms such as Canny and Lin's OPP [9, 38]. Recall that our approach to edge construction borrows some basic ideas and techniques from numerical continuation methods [24]. Continuation methods are used to trace the roots of the expression $G(y, \lambda) = 0$ as the parameter λ is varied. Since the OPP freeways are a subset of the GVG (Section 4.7.1), the OPP uses the same expression, $G_1(y, \lambda) = 0$, to trace its freeway edges.

Recall from Section 4.1 that the robot begins incrementally constructing the GVG, by taking a small step in the direction of the tangent of the GVG edge. This step is the prediction step. The prediction step for the OPP method is always fixed.

Normally, this prediction step takes the robot off of the roadmap, so on a hyperplane orthogonal to the tangent, the robot invokes a correction method to bring itself back to the roadmap. Let $\nabla_y G$ be the matrix formed by taking the derivative of G with respect to coordinates of a hyperplane orthogonal to the step direction. The hyperplane orthogonal to the tangent and passing through x (the starting point of the robot) is termed the normal slice plane, and the correcting slice plane, which is parallel to the normal slice plane, is where the correction procedure takes place.

When $\nabla_y G(y, \lambda)$ is full rank at $x = (y, \lambda)$, it is possible to use an iterative Newton's Method to implement the corrector step. If y^k and λ^k are the k^{th} estimates of y and λ, the $k+1^{st}$ iteration is defined as

$$y^{k+1} = y^k - \left(\nabla_y G\right)^{-1} G(y^k, \lambda^k) \tag{4.24}$$

where $\nabla_y G$ is evaluated at (y^k, λ^k).

In Section 4.1, we demonstrate that $\nabla_y G(y, \lambda)$ is full rank for the GVG. The second part of this section is dedicated to showing $\nabla_y G(y, \lambda)$ is also full rank for the OPP.

Finally, there are two terminating conditions: (1) when the robot encounters the boundary of the environment, and (2) when the robot detects a single object distance gradient is orthogonal to a slice, thereby detecting a fold bifurcation.

4.7.1 OPP is a Subset of GVG

Proposition 4.11 *The freeways (ridge curves) of the OPP method are a subset of the GVG edges.*

Proof: Recall Proposition B.18. It states that at a point $x \in \mathbb{R}^m$, if zero is in the interior of the generalized gradient of the multi-object distance function, D, then D attains a local maxima at x. That is, for $x_* = (\lambda, y_*^T)^T \in \mathbb{R}^m$, if $0 \in \text{int}(\partial_y D(y_*; \lambda)) = \text{int}(\pi_y \partial D(y_*; \lambda))$, then y_* is a local maximum on a slice. This proof hinges on the above proposition, and the following two lemmas whose proofs appear in Section A.3.

Lemma A.21 *Given a set of n arbitrary vectors in $\mathbb{R}^m$, then* $0 \in \text{int}(\text{Co}\{v_i \in$

$\mathbb{R}^m : i = 1, \ldots, n\}$) *if and only if* $\{v_i \in \mathbb{R}^m : i = 1, \ldots, n\}$ *positively span* $\mathbb{R}^m$.

Lemma A.22 *It requires a minimum of* $(m+1)$ *vectors to positively span* $\mathbb{R}^m$.

In the OPP method, the freeways are the trace of local maxima of the multi-object distance function, D, on slices which are diffeomorphic to $\mathbb{R}^{m-1}$. Proposition B.18 indicates that for all points on the freeway, the origin must be in the generalized gradient of the multi-object distance function, ∂D, constrained to a slice. Recall that x is equidistant to h obstacles, if and only if the generalized gradient of D is the convex hull of h single object distance gradient vectors.

By Lemma A.21, the origin is in the convex hull of a collection of vectors if and only if those vectors positively span the space. Therefore, by Lemma A.21 and Proposition B.18, if x is a local maxima of D, constrained to a slice, then the single object distance gradient vectors positively span $T_x\mathbb{R}^{m-1}$ which is diffeomorphic to $\mathbb{R}^m$. Lemma A.22 guarantees that the generalized gradient of D is the convex hull of at least m gradient vectors. Therefore, x is equidistant to at least m obstacles at a local maxima of D.

When x is a local maxima of D, not only is it equidistant to m objects, but by definition of D, the point x is closer to these m objects than to any other object. Thus, the OPP freeway segments are a subset of the GVG edges. ■

Figure 4.16 depicts the correspondence between local maxima of D and equidistance of points in the GVG to nearby obstacles.

4.7.2 Traceability

Since the OPP freeway components are a subset of the GVG edges, the OPP freeway edges are generated by tracing the roots of the same function, G_1, whose roots define the GVG. Recall from Section 4.1 that the roots of G_1 can be found via a two step procedure: (1) a prediction stage and (2) a correction stage. The GVG prediction step occurs along the tangent to a GVG edge, whereas the OPP prediction step occurs in a fixed "sweep" direction. The correction phase requires that ∇G be invertible, which is shown to be true on a hyperplane orthogonal to a GVG edge.

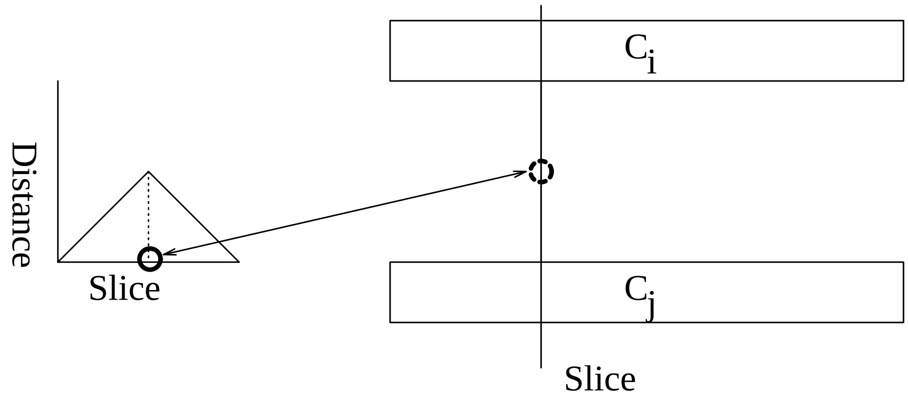

Figure 4.16 The local maxima of the multi-object distance function occurs only at equidistance. The two circles both correspond to the same point. The dashed circle surrounds the point, equidistant to objects C_i and C_j. The solid circle surrounds the point on the slice where distance is maximized, as depicted by the graph of the distance function in the left-hand side.

The following proposition guarantees that ∇G is invertible during the correction phase for the OPP.

Proposition 4.12 *In a neighborhood of a point x on a freeway, $\nabla_Y G(x)$ is invertible.*

Proof: It will be shown that ∇G is invertible on a freeway, and then by continuity of the distance function and the determinant function, $\nabla G(x)$ is invertible in a neighborhood of the OPP freeways. This proof relies on the following lemma whose proof appears in Section A.3

Lemma A.23 *If $\{v_1, \ldots, v_m\}$ postively span $\mathbb{R}^{m-1}$, then $\{v_1-v_2, v_1-v_3, \ldots, v_1-v_m\}$ span $\mathbb{R}^{m-1}$.*

Lemma 4.13 *At a point x on a freeway, $\nabla_Y G(x)$ is invertible.*

Proof: Let $x = (\lambda, y^T)^T$ be the coordinates where the y coordinates span the hyperplane, Y, orthogonal to the sweep direction and λ corresponds to the sweep direction. Proposition B.18 states that if $0 \in \mathrm{int}(\partial_Y D(y;\lambda)) = \mathrm{int}(\pi_Y \partial D(y;\lambda))$, then y is a local maximum. Therefore, if $x = (\lambda, y^T)^T$ is a point on a freeway, then $0 \in \mathrm{Co}\{\nabla_Y d_1, \ldots, \nabla_Y d_m\}$.

By Lemma A.21, $0 \in co\{\nabla_Y d_1, \ldots, \nabla_Y d_m\}$ if and only if the vectors $\{\nabla_Y d_1, \ldots, \nabla_Y d_m\}$ positively span the slice. Therefore, at a local maxima (i.e., a point on a freeway), $\{\nabla_Y d_1, \ldots, \nabla_Y d_m\}$ positively span the slice.

Finally, by Lemma A.23, since $\{\nabla_Y d_1, \ldots, \nabla_Y d_m\}$ positively span the slice, $\{\nabla_Y d_1 - \nabla_Y d_2, \nabla_Y d_1 - \nabla_Y d_3, \ldots, \nabla_Y d_1 - \nabla d_m\}$ span the slice. That is, they are linearly independent.

Therefore,

$$\nabla_Y G(y, \lambda) = \begin{bmatrix} \nabla_Y d_1(y,\lambda) - \nabla_Y d_2(y,\lambda) \\ \nabla_Y d_1(y,\lambda) - \nabla_Y d_3(y,\lambda) \\ \vdots \\ \nabla_Y d_1(y,\lambda) - \nabla_Y d_m(y,\lambda) \end{bmatrix}$$

is invertible. ▼

Since the determinant function and distance function are continuous, there exists a neighborhood about the OPP freeways for which $\nabla_Y G$ is invertible. ■

4.8 Conclusion

This chapter introduced an incremental procedure to construct the GVG and the HGVG. This procedure requires only local sensor distance measurement data, and is therefore practically implementable, as demonstrated by our simulations. Hence, the generalized Voronoi graph and hierarchical generalized Voronoi graph introduced in this work appear to be useful means for implementing sensor based motion planning algorithms. We also believe that with small modifications, some of the numerical methods introduced here can also be useful for "sensorizing" other (e.g., the OPP method) robot motion planners. The next step is to demonstrate the validity of these approaches on an actual mobile robot.

Chapter 5

Implementation and Experiments

5.1 The Robot

The robot used in our experiments is a “B12 Mobile Robot Base” (B12) produced by Real World Interface, Inc.[1] (also described in [22]). The robot's base has three synchronously driven wheels which remain parallel at all times. The robot contains an on-board Motorola 68000 computer that directly controls the motion of the B12. The robot also has a “G96 Sonar Board,” which controls a ring of twelve ultrasonic sensors, each radially pointing outward and equally distributed around the circumference of the robot. See Figure 5.1.

Dead reckoning of the robot's translation is accomplished by integrating the number of encoder counts produced by wheel rotation. Net rotation is measured by a sensor which measures the roation of the base. This procedure does not take into consideration slippage of the robot's wheels.

5.2 Sensor Model

The B12 mobile robot uses ultrasonic sensors to measure distance to environmental features. These sensors determine distance by measuring the time of flight of the ultrasound pulses that reflect off an object and return to the sensor. Although these sensors provide accurate distance measurements, their azimuth readings are not precise. For this reason, we develop a simple sonar sensor model that is com-

[1] Real World Interface, Inc., P.O. Box 270, Main Street, Dublin, NH 03444 USA

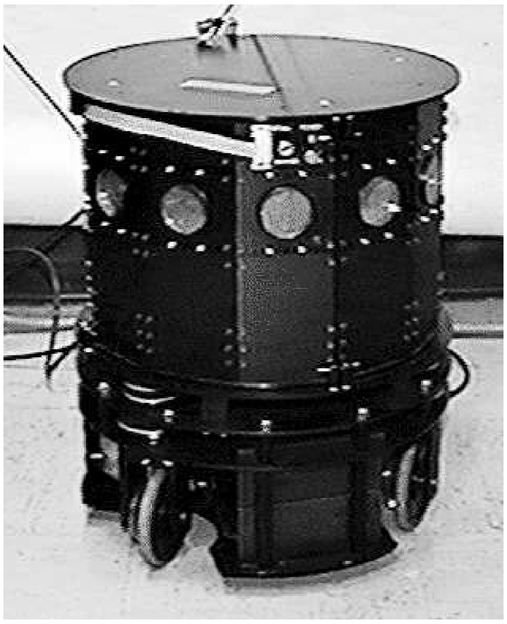

Figure 5.1 B12 Robot.

patible with the incremental construction procedure of the GVG. We assume that the sensors are rigidly attached, pointing radially outward from the robot. The sensors measure distance to nearby obstacles, along a fixed direction termed the *sensor measurement axis.* The absolution orientation of the sensor measurement axis is a function of the robot's position and orientation (See Figure 5.2). Finally, the distance gradient associated with a particular sensor is assumed to be a unit vector pointing along the sensor measurement axis away from the robot. Since the closest point may occur anywhere within the sensor beam pattern, and it is assumed that the distance gradient points along the beam pattern centerline; this can induce errors in the direction of the gradient. However, the accumulated error decreases as the number of sensors increases.

5.3 Distance Function

Incremental construction of the GVG is based on the distance function, d_i, the distance to the nearest point on object C_i. Sensors provide the distance to the nearest

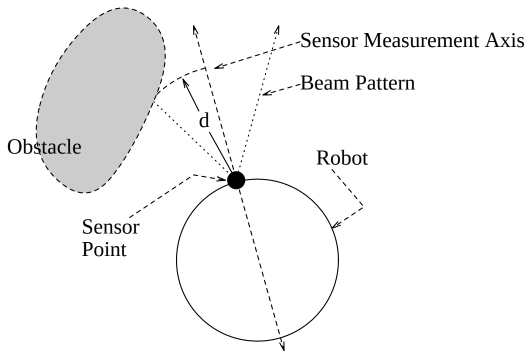

Figure 5.2 Simplified Distance Measurement Sensor Model.

point from the sensor, without knowing from which obstacle the nearest point came. Therefore, the incremental construction of the GVG described in Chapter 4 and in [17] has to be adapted for sensor based implementation on actual robots. In order to do this, we recall the raw distance function which was defined in Section 4.4.

5.3.1 Raw Distance Function

A key feature of the generalized Voronoi graph, and related structures, is that they are defined in terms of a distance function, a function that encodes distance between the robot and nearby obstacles. This makes the GVG and HGVG well suited for sensor based use because many sensors provide range information. The *raw distance function*,

$$\rho(x,s) = \|x + \lambda s\| \quad \text{where } \lambda = \min_{\Lambda \in [0,\infty)} D(x + \Lambda s) = 0, \tag{5.1}$$

provides the distance to all the points on the boundary of the environment that are *within line of sight* of the robot. See Figure 4.5 for an example of the raw distance function.

A key feature of the raw distance function is that it can be readily approximated by many realistic sensor configurations. The sensor measurement provides an approximate value of the distance function $\rho(x,s)$, and the direction to which the sensor is facing corresponds to the direction of measurement ($s \in S^{m-1}$). See

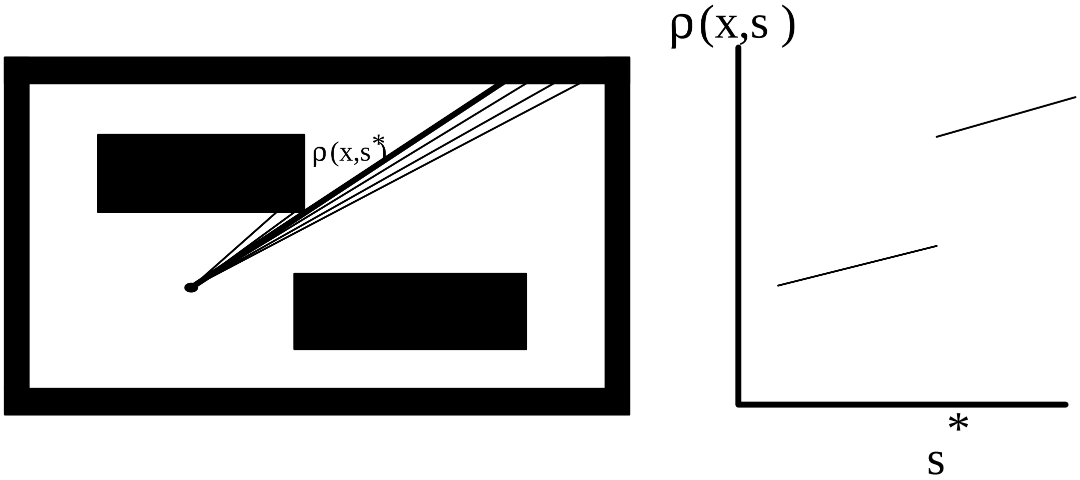

Figure 5.3 The solid polygons are obstacles, and the rays emanating from the point x are values of $\rho(x,s)$ for a fixed x and $s \in \mathrm{nbhd}(s^*)$. As s passes s^*, there is a jump in the value $\rho(x,s)$.

Figure 5.1. In Figure 5.3, it can be seen that the raw distance function is not continuous. At a point $x \in \mathbb{R}^m$, $\lim_{s\to s^{*+}} \rho(x,s) \neq \lim_{s\to s^{*-}} \rho(x,s)$, for $s \in S^{m-1}$.

Definition 5.1 (Cone of Continuity) The *cone of continuity*, $\varrho(x)$ at a point x, is the closure of the set of directions for which $\rho(x,s)$ is continuous with respect to $s \in S^{m-1}$.

Note that $\varrho(x) \subset S^{m-1}$. Typically, at a point x there is more than one cone of continuity, so we attach an index, α, to it, and denote ϱ^α as the αth cones of continuity at x. Note that $S^{m-1} = \bigcup_\alpha \varrho^\alpha(x)$.

In the planar case the cone of continuity is simply a closed interval. See Figure 5.4.

5.3.2 Compute Distance with Raw Distance Function

We can now relate the raw distance function to the slanted and visible distance functions, defined in Section 2.1. See Figure 2.7 for an example of the visible distance function.

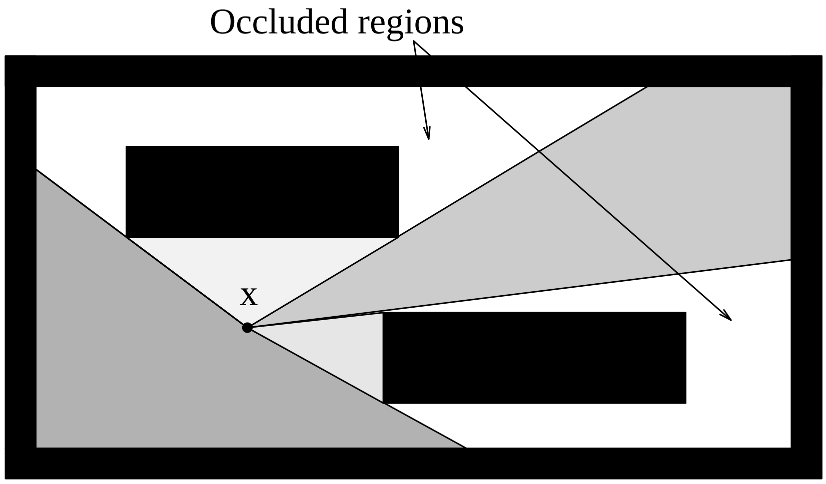

Figure 5.4 The differently shaded regions each correspond to a cone of continuity. The unshaded region corresponds to the points that are not within line of sight of x.

Lemma 5.2 *The visible distance (V-distance) to an object at a point x is a local minima of the raw distance function in the interior of a cone of continuity.*

Proof: Recall the definition of the visible distance function:

$$d_i^V(x) = \begin{cases} \min_{c \in \tilde{C}_i(x)} \|x - c\|, & \text{if } c \in \operatorname{int}(\tilde{C}_i(x)), \\ \infty, & \text{if } c \notin \operatorname{int}(\tilde{C}_i(x)), \end{cases} \tag{5.2}$$

where $\tilde{C}_i(x)$ is the set of points on an object C_i that are within line of sight of x, i.e.,

$$\tilde{C}_i(x) = \{c \in \partial C_i : \forall t \in [0,1],\, x(1-t) + ct \in \mathcal{FS}\}.$$

The αth cone of continuity, $\varrho^\alpha(x)$, can be broken down into sub-regions, each associated with a particular obstacle. The set directions associated with only the boundary of C_i (i.e., points in $\operatorname{int}(\tilde{C}_i(x))$) be denoted $\tilde{\varrho}_i^\alpha(x)$. Clearly, $\bigcup_{i \in I(x)} \tilde{\varrho}_i^\alpha(x) = \varrho^\alpha(x)$ where the index set $I(x)$ corresponds to each obstacle associated with the cone of continuity.

Since $c = x + \rho(x,s)s$, another way to represent $\tilde{C}_i(x)$ is

$$\tilde{C}_i(x) = \{x + \rho(x,s)s \in \partial C_i : \forall s \in \varrho_i^\alpha(x)\}.$$

And since $\rho(x,s) = \|x - c\|$, $c = x + \rho(x,s)s = x + \|x - c\|s$ which implies that $s = \frac{x-c}{\|x-c\|}$. Therefore, there is a one to one correspondence between each $s \in \varrho_i^\alpha(x)$ and $c \in \tilde{C}_i(x)$. Hence,

$$\begin{aligned} d_i^V(x) &= \min_{c \in \mathrm{int}(\tilde{C}_i(x))} \|x - c\| \\ &= \min_{x+\rho(x,s)s \in \mathrm{int}(\tilde{C}_i(x))} \|x - x - \rho(x,s)s\| \\ &= \min_{x+\rho(x,s)s \in \mathrm{int}(\tilde{C}_i(x))} \|\rho(x,s)s\| \\ &= \min_{x+\rho(x,s)s \in \mathrm{int}(\tilde{C}_i(x))} \rho(x,s) \\ &= \min_{s \in \mathrm{int}(\tilde{\varrho}_i^\alpha(x))} \rho(x,s). \end{aligned}$$

■

Lemma 5.3 The slanted distance (S-distance) to an object at a point x is a local minima of the raw distance function in a cone of continuity.

Proof: The proof of this Lemma parallels that of the proof of Lemma 5.2. ■

Since the slanted distance function and the visible distance function are the same for all points on the GVG, the distances to the m closest obstacles which define a GVG edge are the values of the m smallest local minima of the piecewise continuous regions of the raw distance function.

5.3.3 Distance Function

The robot must be able to convert raw sensor readings into distance function readings while ideally avoiding a costly obstacle segmentation procedure in order to effectively perform the incremental construction procedure.

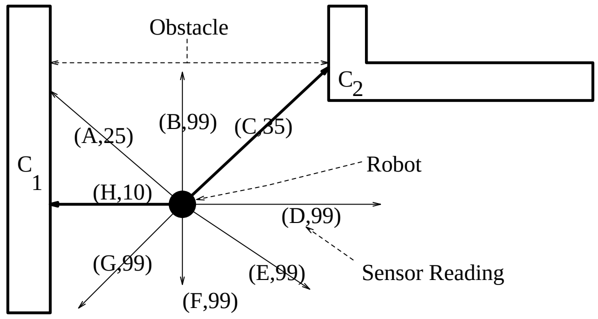

Figure 5.5 Sensor Readings

It is not sufficient to sample the two smallest sensor readings to determine the distance to the two closest obstacles because multiple sensors may detect the same obstacle. The minimum distance to each of the obstacles can be found from the local minima in the circular array sensor readings. An example is depicted in Figure 5.5 where a robot with eight sensors and their measurements is drawn. Sensor H has the smallest value, 10, and is thus pointing at the nearest obstacle. Altough Sensor A has the second smallest measurement, it is not associated with the second closest obstacle because it is not a local minima. Sensor C is associated with the second closest obstacles because its value is the second smallest local minimum in the sensor array. The distance gradients are the unit vectors pointing along the respective sensor centerlines. This method bypasses a costly obstacle segmentation procedure and enables construction of the GVG directly from range sensor data.

5.4 Implementation on a Mobile Robot

Once the distance to individual convex obstacles can be identified, the robot can generate a GVG using the procedures set forth in Chapter 4.

Incremental accessibility is simply gradient ascent applied to the distance to the nearest obstacle. Since the nearest obstacle is associated with the sensor reading with the smallest value, simply moving in a direction opposite to which the sensor with the smallest value is facing is gradient ascent of the distance to the nearest obstacle.

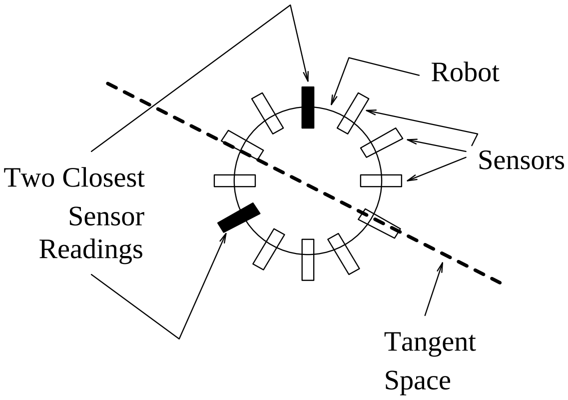

Figure 5.6 The tangent space is defined by the closest sensor readings.

Once the robot has found a GVG edge, it must **incrementally trace** the edge and store an internal representation of it. The distance measurement method, described in Section 5.3.3, determines the distance and direction of the two closest obstacles.

Since there is a finite number of sensors, the robot can rely upon a lookup table scheme to determine the tangent space of the GVG. The orientation of the tangent spaces corresponding to each closest sensor pair is stored in the lookup table, indexed by the two closest sensor locations. Since there are only twelve sensors, a lookup table is a good trade-off between speed and memory storage. The robot orients itself onto the tangent space of the GVG edge and then takes a fixed step along the GVG edge's tangent direction.

In the current implementation, the correction step is an adaptation of the procedure described by Equation (4.3), which prescribes the direction and magnitude of the robot's correction course. Upon completion of the prediction step, the robot makes a ninety degree turn which points the robot in the direction prescribed by Equation (4.3). Instead of moving by the amount specified in Equation (4.3), the robot rolls in a straight line until its two smallest sensor readings are equivalent within a threshold. After reaching the GVG, the robot repeats the step-correct procedure until it encounters a meet point or a boundary point.

It is worth noting that the sensor associated with the nearest reading to an ob-

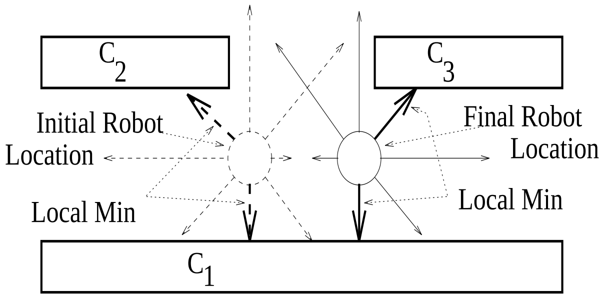

Figure 5.7 Meet Point.

stacle will change in a continuous fashion as the robot explores a GVG edge. When there is an "abrupt" change in a sensor associated with one of the two closest obstacles, then the robot has passed by a *meet point*. An "abrupt" change is a function of the resolution of the sensor system and robot's step size. In our experiments, since we have very few sensors and thus a very low resolution, an "abrupt" change was indicated by a shift of the local minimum by one sensor location (See Fig. 5.7).

Finally, when the robot enters a corner (i.e., reaches a boundary point), it simply turns around and retraces its steps to a previous meet point with unexplored GVG edges.

Another important feature of the GVG is that it produces a concise representation of the world. The GVG edge is stored as a list of points and the GVG vertices are stored as list of pointers, each pointing to a GVG edge with which the GVG vertex is associated. This representation is a significant savings in storage when compared to other world models such as a discrete pixel representation. Furthermore, in tracing out the GVG edges, there is no need to fully identify each obstacle (i.e., perform complicated obstacle segmentation) and thus there is no need to store a representation of each obstacle.

Since the goal of this research is the incremental construction of an environmental map that can be used to make future excursions into an environment, we did not implement the incremental departability algorithm.

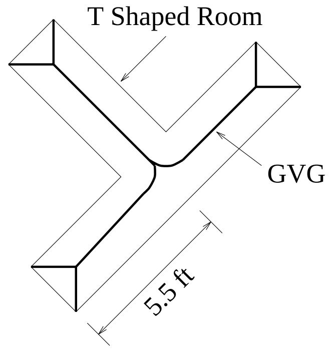

Figure 5.8 T-shaped Room with Actual GVG.

5.5 Experimental Results

The results of one experiment is shown in Figs. 5.8 and 5.9. In this trial, the room was "T-shaped," with the geometry of the room and the theoretical GVG shown in Fig. 5.8. The experimental GVG constructed by the robot is shown in Fig. 5.9. The small squares denote the edge termination points, while the hatched squares represent meet points. For safety reasons, the robot does not trace the edge all the way to the wall's boundary. The octagon shown on the graph represents the point where the robot first accessed the GVG. The experimental GVG edges are jagged because the tangent is crudely approximated because of the angular inaccuracy of sonar distance sensors and the low resolution of sensor placement. However, the GVG is connected, and the edges are far away from the workspace boundary. Our experiments show that the actual GVG construction is quite robust even with crude distance sensors having large errors in distance measurements.

Two more experiments were performed. One of them is simply a hallway (see Figures 5.10 and 5.11 for displayed result) and the other is an "L-shaped" room (see Figs. 5.12 and 5.13 for displayed result).

5.6 Conclusion

This chapter described the implementation of a general sensor based planning strategy, based on the generalized Voronoi graph, for the special case of a planar environ-

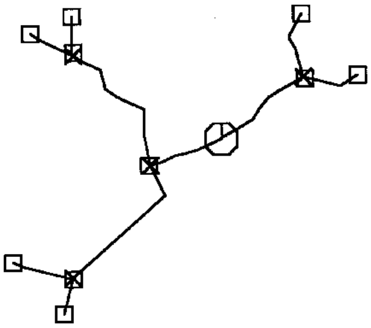

Figure 5.9 Experimental GVG.

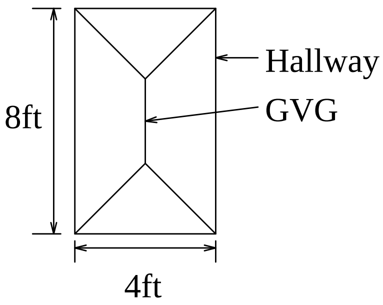

Figure 5.10 Hallway with Actual GVG.

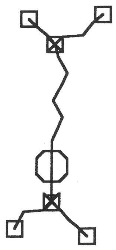

Figure 5.11 Experimental GVG.

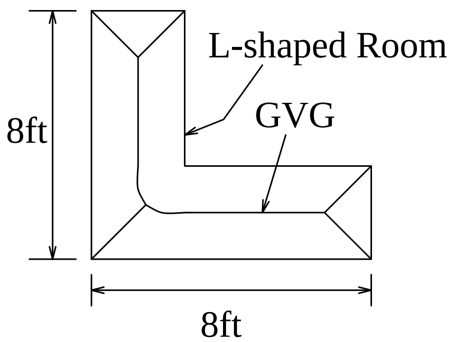

Figure 5.12 L-shaped Room with Actual GVG.

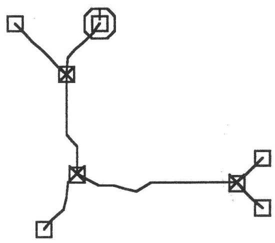

Figure 5.13 Experimental GVG.

ment. We showed that using the algorithm of Chapter 4, a robot equipped with only a ring of sonar sensors can explore an a priori unknown environment and produce a one-dimensional representation (the GVG) of that static environment.

With this one-dimensional representation, the robot can plan future excursions into the environment. One avenue of current work includes using the GVG to plan a shortest distance path between two points. First, a path is determined by planning a path to the GVG, then along the GVG and finally to the goal. Next, this path is deformed so that its length is minimized without violating any obstacle constraints.

In this work, we did not consider all of the implications of sensor noise, and limited sensor range on our algorithm. Also, we assume that the robot has a relatively accurate dead reckoning system. The next step in our research is to consider the issues of sensor noise, limited sensor range, limited sensor resolution and dead reckoning for the planar robot. It is our belief that there are features embedded in the GVG structure which lend themselves to the issues of limited sensor range, sensor noise, and dead reckoning.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Sensor based planning incorporates sensor information, reflecting the current state of the environment, into a robot's planning process, as opposed to classical planning which assumes full knowledge of the world prior to planning. We have developed a rigorous basis for sensor based motion planning for a robot, modeled as a point, in a bounded three-dimensional Euclidean space. Specifically, we defined the *hierarchical generalized Voronoi graph* (HGVG) to serve as a basis for robotic sensor based motion planning. The HGVG is a roadmap is a one-dimensional representation of an environment populated with obstacles, and has three key properties: (1) accessibility, (2) connectivity, and (3) departability.

We make no claim that the HGVG has any clear advantage over other roadmap approaches; however, we demonstrate in a rigorous fashion that the HGVG is sufficient for motion planning. A major advantage that this approach has over other methods is its well defined technique to incrementally construct the HGVG using only local line of sight information. This is necessary for sensor based planning because typically there does not exist one vantage point from which the robot can see the entire environment, so the robot must intelligently traverse the environment in order to build up a complete representation of it. Finally, the incremental construction techniques described in this work can be applied to other motion planning schemes. Simulations and experiments validate this approach.

Finally, in this work we developed techniques for differentiating nonsmooth functions and applied them to the distance function, which is not smooth at its extremal points. Furthermore, we defined and analyzed distance functions which measure distance only to obstacles that are within line of sight of the robot. These distance function results are applicable to other motion planning schemes in addition to the HGVG.

6.2 Future Work

The ultimate goal of work initiated in this thesis is to enable an articulated and arbitrarily shaped robots equipped with sensors to explore and map realistic unknown environments. The HGVG is just a first step towards this goal, but provides a rigorous basis for future work research directions, described below.

6.2.1 Sensor Based Motion Planning for Set Robots

The original problem that motivated this work was sensor based planning for highly articulator robots such as robot snakes [11, 12, 14, 13, 37]. Such robots can be represented as a point in a multi-dimensional configuration space. The analysis done in this work only applies to robots whose configuration spaces are diffeomorphic to $\mathbb{R}^m$ and is thus applicable to mobile robots i the plane, to floating robots that can only translate, and to highly articulated robots with only translational (sliding) degrees of freedom.

The next step would seemingly be to extend the definitions in this work to apply to point robots operating in other types of configuration spaces; however, this may prove to be difficult because there is no clear way to infer configuration space distance information from workspace distance measurements. Instead, we are developing new roadmaps in terms of workspace distance functions, which measure distance between a robot and the environment.

The next evolutionary step is to redefine the HGVG for a robot that can be modeled as a line segment, sometimes called a *rod* (see Figure 6.1). The resulting

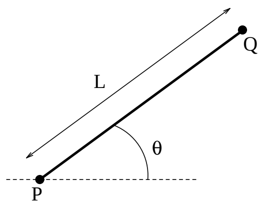

Figure 6.1 The configuration of a rod is determined by the x and y coordinates of P and the orientation of the rod with respect to the horizontal.

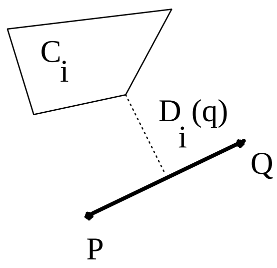

Figure 6.2 The distance from the rod (thick solid line) to an obstacle is the distance (dotted line) between the nearest point on the rod to the obstacle and the nearest point on the obstacle to the rod.

roadmap is termed the *rod hierarchical generalized Voronoi graph* (rod-HGVG), the planar version of which has already been defined in [18]. The rod-HGVG is defined in terms of the *rod single object distance function* which is the shortest distance between a rod R at configuration q and a convex obstacle C_i. See Figure 6.2. The rod distance function is denoted $D_i(q) = \min_{r \in q(R), c \in C_i} \|r - c\|$ where $q(R)$ is the set of points in the plane occupied by a rod, R. Since the rod-HGVG is defined in terms of a distance function, it can be incrementally generated using procedures similar to those described in Section 4.1.

Once the rod-HGVG is finished, its definitions will be extended to the case of the convex set robot. Next, we will define a roadmap for two convex sets, connected by a joint. This will provide the foundation for developing a sensor based motion planning scheme for a chain of convex set robots (i.e., a snake robot).

6.2.2 Sensor Issues and Mobile Robots

Dead reckoning is the process in which a robot determines its location relative to an absolute coordinate system. The HGVG can be further refined to take into account dead reckoning. Throughout this work, it is assumed that the robot knows its location at all times (i.e., it has a GPS-like system on board), which was a satisfactory assumption for our experiments, but is not for large realistic environments. Nevertheless, there are geometries embedded in the HGVG which can help the robot determine its actual position to within some error bound. For example, at each meet point, we can store more information about the environment; this is not memory-consuming because the meet points form a set of measure zero on the HGVG. Each time the robot returns to the vicinity of a meet point, it can use this detailed information to localize itself.

A parallel direction of current research focuses on sensor limitations such as sensor noise and effective sensor range. Recent work considers the development of the *saturated generalized Voronoi graph* which is defined in terms of a distance function that can accurately measure distance only over a pre-specified range. Outside of this range, the distance is considered to be infinite.

Typically, when incrementally constructing the GVG and HGVG, the robot explores edges which are redundant; such redundant GVG edges can be removed from a connected GVG component without affecting the component's connectivity. Therefore, the exploration of such edges can be omitted, thereby decreasing the time needed to explore and map an unknown environment. The resulting structure is termed the *reduced generalized Voronoi graph.* Current work includes using the raw distance function to eliminate redundant edge traces. The raw distance function will also be useful in dealing with sensor quantization. Sensor quantization considers the discretization of range data.

Finally, near-term work will deal with the "two-and-a-half dimensional problem." This problem is part two-dimensional because the robot operates on the surface of a non-flat terrain, but this problem is also part three-dimensional because the robot must decide if hills and valleys should be viewed as obstacles or not.

6.2.3 Robot Vision and Sensor Placement

The incremental construction procedure may require robot vision in environments where range data is not readily available. For example, if the robot is traveling down a sidewalk, there are no walls on the sidewalk's periphery which the robot can use to center itself in the middle of the sidewalk. In this case, the robot must rely on a vision sensor.

Not only can the HGVG draw upon prior work in machine vision, but it also has applications to vision in the field of active perception. In particular, the HGVG can be applied to *the art gallery problem* which deals with the placement of stationary guards in an environment such that the sum of the views of all of the guards covers the entire environment. A variant of this problem is determining the path that a guard must take in order to see the entire environment. The HGVG prescribes such a path.

6.2.4 Injection Molding

It is our hope that the HGVG will have non-robotic applications. One near-term use of the HGVG is in injection molding. Injection molding is a process in which molten plastic is forced into a mold, and then when the plastic hardens, it assumes the shape of the mold.

One of the tradeoffs in part design is manufacturing cost. For injection molding parts, required flow length is the maximal distance plastic has to flow in the mold. Currently, the required flow length is determined late in the design process when most of the part manufacturing cost is already locked into the design. The HGVG can be used to determine the required flow length early in the design process, and thus give more freedom to the designer.

Appendix A

Relevant Mathematical Material

The results contained in this chapter are independent of the assumptions stated in the thesis and apply to any m-dimensional Euclidean vector space.

A.1 Useful Topology Definitions and Relationships

The following is a list of basic topology definitions and results which are useful to this work. Although the following definitions are standard to point set topology, we used the definitions from Marsden's book entitled *Elementary Classical Analysis* [32].

A.1.1 Topology Notation and Definitions

Definition A.1 (ϵ-Neighborhood) ([32], p. 32) An *ϵ-neighborhood* about a point $x \in \mathbb{R}^m$ is the set $\{y \in \mathbb{R}^m : \|x - y\| < \epsilon\}$.

Definition A.2 (Open Sets) ([32], p. 33) A set $A \subset \mathbb{R}^m$ is said to be *open* if for each $x \in A$, there exists a $\epsilon > 0$ such that an ϵ-neighborhood of x is fully contained in A.

Definition A.3 (Neighborhood of a Point) A neighborhood of a point x, denoted $\text{nbhd}(x)$, is an open set containing x.

Definition A.4 (Interior of a Set) ([32], p. 36) For any set $A \subset \mathbb{R}^m$, a point $x \in A$ is an *interior point* of A if there is an open set U such that $x \in U \subset A$. The *interior of* A is the collection of all interior points of A and is denoted $\mathrm{int}(A)$. This set might be empty.

Definition A.5 (Closed Sets) ([32], p. 37) A set B in $\mathbb{R}^m$ is said to be *closed* if its complement in $\mathbb{R}^m$ is open.

Definition A.6 (Accumulation Points) ([32], p. 39) A point $x \in \mathbb{R}^m$ is called an *accumulation point* of a set A if every open set U containing x contains some point of A other than itself.

In a subsequent lemma, we say an accumulation point of a set A is a point x such that in a neighborhood of x, there exist points other than x in A. That is, $(\mathrm{nbhd}(x)\backslash\{x\}) \bigcap A \neq \emptyset$.

Definition A.7 (Closure of a Set) ([32], p. 41) Let $A \subset \mathbb{R}^m$. The set $\mathrm{cl}(A)$ is defined to be the intersection of all closed sets containing A.

Theorem A.8 *([32], p. 42) Let* $A \subset \mathbb{R}^m$*. Then* $\mathrm{cl}(A)$ *consists of* A *plus all of the accumulation points of* A*.*

Definition A.9 (Boundary of a Set) ([32], p. 43) For a given set A in $\mathbb{R}^m$, the *boundary* is defined to be the set

$$\partial A = \mathrm{cl}(A) \bigcap \mathrm{cl}(\mathbb{R}^m \backslash A).$$

A.1.2 Useful Topology Results in Sensor Based Planning

The following proposition is useful for enumerating the boundary components of the structures used to define the HGVG. However, the result is general to all sets in $\mathbb{R}^m$.

Proposition A.10 *For two subsets* A *and* B *of* $\mathbb{R}^m$*, the boundary of the intersection of the two sets obeys the following inclusion:*

$$\partial(A \bigcap B) \subset \left(\partial A \bigcap \mathrm{cl}(B)\right) \bigcup \left(\partial B \bigcap \mathrm{cl}(A)\right).$$

Proof: This proof depends upon the result of the following lemmas.

Lemma A.11 *For two subsets A and B of $\mathbb{R}^m$, the closure of the intersection of A and B is a subset of the intersection of the closures of A and B. In other words,*

$$\mathrm{cl}(A \bigcap B) \subset \mathrm{cl}(A) \bigcap \mathrm{cl}(B).$$

Proof: By Theorem A.8, for all $x \in \mathrm{cl}(A \bigcap B)$, there exists a neighborhood, $Y = \mathrm{nbhd}(x)$, such that $(Y \backslash \{x\}) \bigcap A \bigcap B \neq \emptyset$. Since $A \bigcap B \subset A$, $(Y \backslash \{x\}) \bigcap A \neq \emptyset$. Similarly, $(Y \backslash \{x\}) \bigcap B \neq \emptyset$, and thus all points $x \in \mathrm{cl}(A \bigcap B)$ are accumulation points for both sets A and B. Therefore, for all $x \in \mathrm{cl}(A \bigcap B)$, $x \in \mathrm{cl}(A) \bigcap \mathrm{cl}(B)$. ▼

Lemma A.12 *For two subsets A and B of $\mathbb{R}^m$, the closure of the union of A and B is equal to the union of the closures of A and B. In other words,*

$$\mathrm{cl}(A \bigcup B) = \mathrm{cl}(A) \bigcup \mathrm{cl}(B).$$

Proof: First, we show $\mathrm{cl}(A \bigcup B) \subset \mathrm{cl}(A) \bigcup \mathrm{cl}(B)$. By Theorem A.8, for all $x \in \mathrm{cl}(A \bigcup B)$, there exists a neighborhood, $Y = \mathrm{nbhd}(x)$, such that $(Y \backslash \{x\}) \bigcap (A \bigcup B) \neq \emptyset$. Since $A \subset A \bigcup B$ and $B \subset A \bigcup B$, either $(Y \backslash \{x\}) \bigcap A \neq \emptyset$ or $(Y \backslash \{x\}) \bigcap B \neq \emptyset$. Therefore, all points $x \in \mathrm{cl}(A \bigcup B)$ are accumulation points for either A, B, or both, and thus, for all $x \in \mathrm{cl}(A \bigcup B)$, $x \in \mathrm{cl}(A) \bigcup \mathrm{cl}(B)$.

And now, we show $\mathrm{cl}(A) \bigcup \mathrm{cl}(B) \subset \mathrm{cl}(A \bigcup B)$. By Theorem A.8, if $x \in \mathrm{cl}(A) \bigcup \mathrm{cl}(B)$ then there exists a neighborhood, $Y = \mathrm{nbhd}(x)$, such that $((Y \backslash \{x\}) \bigcap A) \bigcup ((Y \backslash \{x\}) \bigcap B) \neq \emptyset$. In other words, $(Y \backslash \{x\}) \bigcap (A \bigcup B) \neq \emptyset$. Therefore, all points $x \in \mathrm{cl}(A) \bigcup \mathrm{cl}(B)$ are accumulation points of $A \bigcup B$, and thus for all $x \in \mathrm{cl}(A) \bigcup \mathrm{cl}(B)$, $x \in \mathrm{cl}(A \bigcup B)$.

▼

By Defintion A.9,

$$
\begin{aligned}
&\left(\partial A \bigcap \mathrm{cl}(B)\right) \bigcup \left(\partial B \bigcap \mathrm{cl}\,(A)\right) \\
&\quad = \left(\mathrm{cl}\,(A) \bigcap \mathrm{cl}\,(\mathbb{R}^m \backslash A) \bigcap \mathrm{cl}\,(B)\right) \bigcup \left(\mathrm{cl}\,(B) \bigcap \mathrm{cl}\,(\mathbb{R}^m \backslash B) \bigcap \mathrm{cl}\,(A)\right) \\
&\quad = \mathrm{cl}\,(A) \bigcap \mathrm{cl}\,(B) \bigcap \left(\mathrm{cl}\,(\mathbb{R}^m \backslash A) \bigcup \mathrm{cl}\,(\mathbb{R}^m \backslash B)\right).
\end{aligned}
\tag{A.1}
$$

Also, by Defintion A.9,

$$
\partial(A \bigcap B) = \mathrm{cl}(A \bigcap B) \bigcap \mathrm{cl}(\mathbb{R}^m \backslash (A \bigcap B)) \tag{A.2}
$$

$$
= \mathrm{cl}(A \bigcap B) \bigcap \mathrm{cl}(\mathbb{R}^m \backslash A \bigcup \mathbb{R}^m \backslash B). \tag{A.3}
$$

By Lemmas A.11 and A.12, the following are true:

$$
\begin{aligned}
\mathrm{cl}\left(A \bigcap B\right) &\subset \mathrm{cl}\,(A) \bigcap \mathrm{cl}\,(B), \\
\mathrm{cl}\left((\mathbb{R}^m \backslash A) \bigcup (\mathbb{R}^m \backslash B)\right) &\subset \mathrm{cl}\,(\mathbb{R}^m \backslash A) \bigcup \mathrm{cl}\,(\mathbb{R}^m \backslash B).
\end{aligned}
$$

Therefore,

$$
\mathrm{cl}\left(A \bigcap B\right) \bigcap \mathrm{cl}\left((\mathbb{R}^m \backslash A) \bigcup (\mathbb{R}^m \backslash B)\right) \subset \mathrm{cl}\,(A) \bigcap \mathrm{cl}\,(B) \bigcap \left(\mathrm{cl}\,(\mathbb{R}^m \backslash A) \bigcup \mathrm{cl}\,(\mathbb{R}^m \backslash B)\right),
$$

and using the identities from Equations A.1 and A.2,

$$
\partial(A \bigcap B) \subset \left(\partial A \bigcap \mathrm{cl}(B)\right) \bigcup \left(\partial B \bigcap \mathrm{cl}(A)\right).
$$

■

A.2 Convex Functions

Definition A.13 (Convex Hull) The *convex hull* of the set of vectors $\{v_i : i = 1, \ldots, n\}$ is

$$\mathrm{Co}\{v_i : i = 1, \ldots, n\} = \left\{ \sum_{i=1}^{n} \lambda_i v_i : \lambda_i \in \mathbb{R} \text{ such that } \lambda_i > 0 \ \forall i \text{ and } \sum_{i=1}^{n} \lambda_i = 1 \right\}.$$

Definition A.14 (Convex Function) The function $f\colon \mathbb{R}^m \to \mathbb{R}$ is *convex* if for all $x_i, i = 1, \ldots, n$, $\sum_{i=1}^{n} \lambda_i = 1$, and $0 \leq \lambda_i \leq 1$,

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i).$$

Lemma A.15 *Let f be a convex function and U be the convex hull of $\{x_1, \ldots, x_m\}$. For all $x \in U$, $f(x) \leq \max_i f(x_i)$.*

Proof: Pick $x \in U$ where $x = \bar{\lambda}_1 x_1 + \cdots + \bar{\lambda}_m x_m$.

$$\begin{aligned} f(x) &= f(\bar{\lambda}_1 x_1 + \cdots + \bar{\lambda}_m x_m) \\ &\leq \bar{\lambda}_1 f(x_1) + \cdots + \bar{\lambda}_m f(x_m) \\ &\leq \bar{\lambda}_1 \max_i f(x_i) + \cdots + \bar{\lambda}_m \max_i f(x_i) \\ &= \max_i f(x_i) \textstyle\sum_{i=1}^{i=m} \bar{\lambda}_i \\ &= \max_i f(x_i). \end{aligned} \tag{A.4}$$

■

Lemma A.16 *A convex function only has local maxima on the boundary of its domain.*

Proof: Assume there exists an x which is a local maxima of the convex function f. This implies there exists a neighborhood,$Y = \mathrm{nbhd}(x)$, for which $f(x) > f(y)$ for all $y \in Y$. Choose m points y_i from the neighborhood Y. For all i, $f(x) > \max_i f(y_i)$. This contradicts the result of Lemma A.15. ■

Lemma A.17 *For a non-constant convex function $f\colon \mathbb{R}^m \to \mathbb{R}$, if there exists an open set U such that $f(x) = c$ for all $x \in U$, then $f(y) > c$ for all $y \in \mathbb{R}^m \backslash U$.*

Proof: Let x_1 be in U and $y \notin U$. By hypothesis, there exists a nbhd(x_1) such that for all $x \in$ nbhd(x_1) there exists an $\epsilon > 0$ such that

$$\begin{aligned} f((1-\epsilon)x+\epsilon y) &< (1-\epsilon)f(x)+\epsilon f(y) \\ f(x) &< (1-\epsilon)f(x)+\epsilon f(y) \\ f(x)-(1-\epsilon)f(x) &< \epsilon f(y) \\ f(x) &< f(y) \\ f(x_1) &< f(y). \end{aligned} \tag{A.5}$$

■

A.3 Convex Hulls and Positively Spanning Vectors

Lemma A.18 (Nguyen and Goldman & Tucker) *In an m-dimensional vector space, a set of vectors $\{v_i \in \mathbb{R}^m : i = 1,\ldots,n\}$ has at least $m+1$ vectors $\{v_i \in \mathbb{R}^m : i = 1,\ldots,m+1\}$ if and only if*

1. *m of the $m+1$ vectors are linearly independent, and*
2. *a strictly positive sum of the $m+1$ vectors is the zero vector, i.e., $\sum_{i=1}^{i=m+1} \lambda_i v_i = 0$ where $\lambda_i > 0$ for all i.*

See [33, 23] for proof.

Lemma A.19 *A set of n arbitrary vectors $\{v_i \in \mathbb{R}^m : i = 1,\ldots,n\}$ positvely spans $\mathbb{R}^m$ if and only if there exists a v_i such that for all $w \in \mathbb{R}^m$, $\langle w, v_i\rangle > 0$.*

Proof: Let $w = \sum_{i=1}^{n} \alpha_i v_i$ be a nonzero vector and thus, there must be exist an $\alpha_i > 0$.

$$\begin{aligned} 0 &< \langle w, w\rangle \\ &= \langle \textstyle\sum_{i=1}^{n} \alpha_i v_i, w\rangle \\ &= \textstyle\sum_{i=1}^{n} \langle \alpha_i v_i, w\rangle. \end{aligned} \tag{A.6}$$

This implies that there exists an i where $\langle \alpha_i v_i, w\rangle > 0$ and since $\alpha_i > 0$, we can conclude that $\langle v_i, w\rangle > 0$ for at least one i. ■

Lemma A.20 *The origin is contained in the interior of the convex hull of a set of n arbitrary vectors $\{v_i \in \mathbb{R}^m : i = 1, \ldots, n\}$ if and only if there exists a v_i such that for all $w \in \mathbb{R}^m$, $\langle w, v_i \rangle > 0$.*

Proof: Let $w \in \mathbb{R}^m$ be a nonzero vector. Since $0 \in \text{int}(\text{Co}\{v_1, \ldots, v_n\})$, there exists a $Y = \text{nbhd}(0) \subset \text{int}(\text{Co}\{v_1, \ldots, v_n\})$. Chose a postive $s \in \mathbb{R}$ such that $w_s = sw \in Y$. Therefore, $w_s = \sum_{i=1}^n \alpha_i v_i$ where $\alpha_i \in (0, 1)$ because w_s is in the interior of the convex hull of $\{v_1, \ldots, v_n\}$.

$$\begin{aligned} 0 &< \langle w_s, w_s \rangle \\ &= \langle \textstyle\sum_{i=1}^n \alpha_i v_i, w_s \rangle \\ &= \textstyle\sum_{i=1}^n \langle \alpha_i v_i, w_s \rangle. \end{aligned} \tag{A.7}$$

This implies that there exists an i where $\langle \alpha_i v_i, w_s \rangle > 0$ and since $\alpha_i > 0$, we can conclude that $\langle v_i, w_s \rangle > 0$. That is, $\langle v_i, w_s \rangle > 0$ for at least one i. Finally, sicne $w_s = sw$ for positive s, we can conclude that $\langle v_i, w \rangle > 0$. ■

Lemma A.21 *Given a set of n arbitrary vectors in $\mathbb{R}^m$, then $0 \in \text{int}(\text{Co}\{v_i \in \mathbb{R}^m : i = 1, \ldots, n\})$ if and only if $\{v_i \in \mathbb{R}^m : i = 1, \ldots, n\}$ positively span $\mathbb{R}^m$.*

Proof: This proof is a simple consquence of Lemmas A.19 and A.20. ■

Lemma A.22 (Goldmand and Tucker) *It requires a minimum of $(m + 1)$ vectors to positively span $\mathbb{R}^m$.*

Proof: Assume that the vectors $\{v_1, \ldots, v_m\}$ span $\mathbb{R}^m$ (a minimum of m vectors is required to span $\mathbb{R}^m$). For all $w \in \mathbb{R}^m$, $w = \sum_{i=1}^m a_i v_i$ where $a_i \in \mathbb{R}$. Define the vector, v_{m+1}, which is the negated sum of the other v_i's: $v_{m+1} = -v_1 - v_2 - \cdots - v_m$. Let $\kappa = (\max_i |a_i|) + \epsilon$ where $\epsilon > 0$. Then the following holds for all $w \in \mathbb{R}^m$,

$$\begin{aligned} w &= \textstyle\sum_{i=1}^m a_i v_i \\ &= \textstyle\sum_{i=1}^m (a_i + \kappa) v_i + \kappa v_{m+1} \\ &= \textstyle\sum_{i=1}^{m+1} b_i v_i. \end{aligned} \tag{A.8}$$

where for $i = 1, \ldots, m$, $b_i = a_i + \kappa$ and $b_{m+1} = \kappa$. Since $\kappa > 0$ and $\kappa > |a_i|$ for all i, all the b_i's are positive. Therefore, $m+1$ vectors may positively span $\mathbb{R}^m$.

Now, it needs to be shown that a minimum of $m+1$ vectors are required to positively span $\mathbb{R}^m$. Recall $v_{m+1} = -v_1 - v_2 - \cdots - v_m$. Clearly, v_{m+1} is an element of $\mathbb{R}^m$ and it is not positively spanned by the other m vectors $\{v_1, \ldots, v_m\}$. ■

Lemma A.23 *If* $\{v_1, \ldots, v_m\}$ *postively span* $\mathbb{R}^{m-1}$*, then* $\{v_1 - v_2, v_1 - v_3, \ldots, v_1 - v_m\}$ *span* $\mathbb{R}^{m-1}$*.*

Proof: Since $\{v_1, \ldots, v_m\}$ postively span $\mathbb{R}^{m-1}$, there exists a set of nonzero $\{a_i \in \mathbb{R}\}$ such that $0 = \sum_{i=1}^m a_i v_i$ (Lemma A.18). In fact, by Lemma A.18, we can write one of the vectors as a linear combination of the others. Without loss of generality, $v_1 = -\sum_{i=2}^m \frac{a_i}{a_1} v_i$. For $i > 2$, let $b_i = \frac{a_i}{a_1}$. So, $v_1 = -\sum_{i=2}^m b_i v_i$.

$$
\begin{aligned}
\textstyle\sum_{i=2}^m c_i(v_i - v_1) = 0 \quad &\iff \textstyle\sum_{i=2}^m c_i v_i - \sum_{i=2}^m c_i v_1 = 0 \\
&\iff \textstyle\sum_{i=2}^m c_i v_i - v_1 \sum_{i=2}^m c_i = 0 \\
&\iff \textstyle\sum_{i=2}^m c_i v_i + \left(\sum_{i=2}^m b_i v_i\right)\left(\sum_{i=2}^m c_i\right) = 0 \\
&\iff \textstyle\sum_{i=2}^m \left(c_i + \left(\sum_{i=2}^m c_i\right) b_i\right) v_i = 0 \\
&\iff \textstyle\left(\left(\sum_{i=2}^m c_i\right)\left(1 + \left(\sum_{i=2}^m b_i\right)\right)\right) v_i = 0 \\
&\iff \textstyle\sum_{i=2}^m c_i = 0 \quad \text{or} \quad \sum_{i=2}^m b_i = -1
\end{aligned}
\tag{A.9}
$$

$\sum_{i=2}^m b_i = -1$ cannot be true because $\forall i$, $b_i > 0$, Since by Lemma A.18 the set $\{v_2, \ldots v_m\}$ spans $\mathbb{R}^{m-1}$, $\sum_{i=2}^m c_i v_i = 0$ if and only if $c_i = 0$ for all i. Therefore, $\sum_{i=2}^m c_i(v_i - v_1) = 0$ if and only if $c_i = 0$ for all i and thus $\{v_i - v_1 : i = 2, \ldots, m\}$ are linearly independent. These $m-1$ linearly independent vectors span $\mathbb{R}^{m-1}$. ■

Appendix B

Nonsmooth Analysis

B.1 Introduction to the Distance Function and Nonsmooth Analysis

The application of nonsmooth analysis to the Euclidean distance function is important because this function is an integral component of many path planners, in addition to the one described in this work. Prior work in motion planning has not fully considered the important issues of nonsmoothness when this distance function is employed as the basis for motion planning. In this chapter, we show that this function is in fact *always* nonsmooth at the points that are of interest to motion planning. For example, the freeway curves of the OPP method are the set of the local maxima of the nonsmooth Euclidean distance function constrained to a slice (See Section B.3).

B.2 Review of Nonsmooth Analysis

The Euclidean distance function is nonsmooth at many points of interest and does not have a conventional derivative at these points. However, one can build a calculus for such nonsmooth functions from a less restrictive class of assumptions than smoothness: Lipschitzness, regularity, and convexity. We review here some essential results from nonsmooth analysis and develop a few useful results. A more comprehensive treatment of nonsmooth analysis can be found in [19]. Throughout this section, assume that X is a finite dimensional vector space.

Definition B.1 (Lipschitz) A function $f(x)$ is *Lipschitz near* x when

$$\|f(x_i) - f(x_j)\| \leq K\|x_i - x_j\| \quad \forall x_i, x_j \in \text{nbhd}(x),$$

where K is a positive scalar.

Definition B.2 (Generalized Directional Derivative) The generalized directional derivative *([19], p. 10) of* $f(x)$ *in the direction* v *is*

$$f^\circ(x;v) = \limsup_{y \to x, t \to 0^+} \frac{f(y+tv) - f(y)}{t},$$

for all $t \in \mathbb{R}$, and $y, v \in X$.

Definition B.3 (Regular) The function f is *regular* ([19], p. 39) at x when: (1) for all $v, f^{'}(x,v)$ exists, where $f^{'}$ is the usual one sided derivative; and (2) for all $v, f^{'}(x,v) = f^\circ(x,v)$.

All convex functions are regular ([19], p. 40). It can be shown that the single object distance function of a convex set is a convex function ([19], p. 40).

While a Lipschitz function need not be smooth, it does possess a *generalized gradient*. This definition is key to the notion of a gradient of a nonsmooth distance function.

Definition B.4 (Generalized Gradient) *In a finite dimensional space,* the generalized gradient *([19], p63) of a Lipschitz function* f *at a point* x *is denoted by* $\partial f(x)$ *and given by:*

$$\partial f(x) = Co\{\lim_{x_i \to x} \nabla f(x_i) : x_i \notin S,\ x_i \notin \Omega_f\},$$

where Ω_f *is the set of points where* f *fails to be differentiable,* S *is any set of measure zero, and* Co *means convex hull.*

Note that if $f(x)$ is smooth at x, then $\partial f(x)$ reduces to the conventional gradient. The use of the ∂ can be confusing because it can also mean "boundary" when it precedes a variable that represents a set.

Now, we introduce some properties that are useful in the context of generalized gradients.

Proposition B.5 The negated convex hull of a set of vectors is the convex hull of the negated vectors, i.e.,

$$-\mathrm{Co}\{x_1, \ldots, x_n\} = \mathrm{Co}\{-x_1, \ldots, -x_n\}. \tag{B.1}$$

Proof:

$$\begin{aligned} -\mathrm{Co}\Big\{x_1, ..., x_n\Big\} &= -\{\textstyle\sum_{i=1}^n \lambda_i x_i\} \\ &= \{\textstyle\sum_{i=1}^n -\lambda_i x_i\} \\ &= \{\textstyle\sum_{i=1}^n \lambda_i(-x_i)\} \\ &= \mathrm{Co}\Big\{-x_1, \cdots, -x_n\Big\} \end{aligned}$$

■

Proposition B.6 *([19], p38)* $\partial(sf)(x) = s\partial f(x) \quad \forall s \in \mathbb{R}$.

In particular, note that $\partial(-f) = -\partial(f)$.

Proposition B.7 *([19],p47)* Let $\{f_i\}$, $i = 1, \ldots, n$, be a set of functions which are Lipschitz (respectively regular) near x_0. For x in a neighborhood of x_0, the function

$$f(x) = \max_{i=1,\ldots,n} \{f_i(x)\} \tag{B.2}$$

is also Lipschitz (respectively regular).

Since $f(x)$ in Equation B.2 is Lipschitz, we can define its generalized gradient.

Proposition B.8 (Pointwise Maxima) *([19],p47)* For the function

$$f(x) = \max_{i=1,\ldots,n} \{f_i(x)\}$$

, where each $f_i(x)$ is regular, $f(x)$ is regular and has the generalized gradient,

$$\partial f(x) = \mathrm{Co}\{\partial f_i(x) : i \in I(x)\}, \tag{B.3}$$

where $I(x)$ is the set of indices for which $f_i(x) = f(x)$. That is, $f_i(x)$ attains a maximum for all $i \in I(x)$.

Proposition B.9 (Pointwise Minima) For a set of regular functions $f_i(x)$, the function

$$g(x) = \min_{i=1,\ldots,n}(f_i(x)) = -\max_{i=1,\ldots,n}(-f_i(x)) \tag{B.4}$$

has a generalized gradient given by

$$\partial g(x) = \mathrm{Co}\{\partial f_i(x) \quad \forall i \in I(x)\}, \tag{B.5}$$

where $I(x)$ is the set of indices for which $f_i(x) = g(x)$, i.e., where $f_i(x)$ attains the minimum over all $i \in I(x)$. As before, g is regular.

Proof: The proof is a simple consequence of Proposition B.5 and B.6.

$$\begin{aligned} \partial g(x) &= -\mathrm{Co}\{\partial(-f_i(x)) \ \forall i \in I(x)\} \\ &= -\mathrm{Co}\{-\partial(f_i(x)) \ \forall i \in I(x)\} \\ &= \mathrm{Co}\{\partial f_i(x) \ \forall i \in I(x)\}. \end{aligned}$$

■

B.3 Distance Function

A function which encodes the distance between a robot and nearby obstacles is key to our definitions. This section exploits the properties of the X-distance function and its gradient. These properties are generalizable to the V-distance function. Again, we assume a point robot operating in a subset, termed the work space ($\mathcal{W}$), of an m-dimensional Euclidean space. The work space $\mathcal{W}$ is populated by obstacles

$C_1, \ldots, C_n$ which are convex sets. Non-convex obstacles are modeled as the union of convex shapes. It is assumed that the boundary of $\mathcal{W}$ is a collection of convex sets, which are members of the obstacle set $\{C_i\}$.

Recall the following definitions:

- single object distance function (Definition 2.1)

$$d_i(x) = \min_{c_0 \in C_i} \|x - c_0\|,$$

- single object distance function gradient (Equation 2.2),

$$\nabla d_i(x) = \frac{x - c_0}{\|x - c_0\|},$$

  for $c_0 \in C_i$ such that $\|x - c_0\| = \min_{c \in C_i} \|x - c\|$.

- multi-object distance function (Definition 2.2)

$$D(x) = \min_i d_i(x).$$

B.3.1 Properties of the Distance Function

Proposition B.10 The single object distance function, d_i, is continuous.

Proof: The single object distance function d_i is said to be continuous if for every $\epsilon > 0$, there exists a δ such that $|x - y| < \delta$ implies that $|d_i(x) - d_i(y)| < \epsilon$, where $x, y \in \mathcal{W}$. Let $\delta = \epsilon$ and assume $|x - y| < \delta$.

If $d_i(x) \geq d_j(x)$, we then have

$$\begin{aligned}
|d_i(x) - d_i(y)| &= ||x - c_x| - |y - c_y|| && (\text{where } c_x = \text{argmin } d_i(x) \text{ and } c_y = \text{argmin } d_i(y)) \\
&= |x - c_x| - |y - c_y| && (\text{if } d_i(x) \geq d_j(x)) \\
&\leq |x - c_y| - |y - c_y| && \text{because } |x - c_x| \leq |x - c_y| \\
&\leq |x - y| \\
&\leq \delta.
\end{aligned}$$

Otherwise, if $d_j(x) \geq d_i(x)$, we then have

$$\begin{aligned}
|d_i(x) - d_i(y)| &= ||x - c_x| - |y - c_y|| && \text{(where } c_x = \text{argmin } d_i(x) \text{ and } c_y = \text{argmin } d_i(y)\text{)} \\
&= |y - c_y| - |x - c_x| && \text{(if } d_i(x) \leq d_j(x)\text{)} \\
&\leq |y - c_x| - |x - c_x| && \text{(because } |y - c_y| \leq |y - c_x|\text{)} \\
&\leq |y - x| = |x - y| \\
&\leq \delta.
\end{aligned}$$

Since $\delta = \epsilon$, $|x - y| \leq \delta$ implies that $|d_i(x) - d_i(y)| \leq \epsilon$. ■

Proposition B.11 *([19], p. 51)* The single object distance function, d_i, is Lipschitz.

As a result of Proposition B.7, we have

Proposition B.12 The multi-object distance function, D, is Lipschitz.

B.3.2 Generalized Gradient of the Multi-Object Distance Function

The generalized gradient of the multi-object distance function exists because the multi-object distance function is Lipschitz by Proposition B.12. Since the multi-object distance function is of the form in Equation B.4, where each d_i is regular, Proposition B.9 states that its generalized gradient will have the form

$$\partial D(x) = \text{Co}\{\nabla d_i(x) : i \in I(x)\}. \tag{B.6}$$

Recall that $I(x)$ is the set of indices where $d_i(x) = D(x)$. The physical intuition is:

1. If there is a unique closest point (and hence a unique closest obstacle), $\partial D(x)$ is a unit vector pointing away from the closest point. In this case, $D(x)$ is smooth.

2. If there are a set of equidistant closest points, then $\partial D(x)$ is the convex hull of the gradients with respect to each point. In this case, a robot is equidistant

to multiple convex obstacles.

We now consider in more detail how to compute the gradient of d_i. When there is a *unique* closest point $c_0 \in C_i$ to $x \notin cl(C_i)$, then the gradient is

$$\nabla d_i(x) = \frac{x - c_0}{\|x - c_0\|}. \tag{B.7}$$

That is, ∇d_i is a unit vector emanating from x pointing away from c_0. When C_i is convex, d_i is smooth everywhere but on the boundary of C_i.

However, d_i *is not smooth* when there are multiple closest points. Such a case occurs when C_i is not convex, and x is equidistant to multiple points, $\{c_j\}$, on the boundary of C_i. The single object distance gradient is not smooth and takes the form

$$\partial d_i(x) = \mathrm{Co}\left\{\frac{x - c_j}{\|x - c_j\|} : \quad \forall c_j\right\}. \tag{B.8}$$

However, if a concave object is treated as the union of convex shapes, then the generalized gradient, $\partial D(x)$, is still the same.

Proposition B.13 *The generalized gradient of the multi-object distance function is independent of whether or not concave shapes are decomposed into convex components.*

Proof: In this proof, obstacles may be concave. Let x be a point which is equidistant to k closest points on the boundaries of both concave and convex obstacles, where $k < m$ and m is the dimension of the space. Let K be the number of obstacles to which x is equidistant. That is $d_{i_1}(x) = \cdots = d_{i_K}(x)$. Naturally, $K \leq k$ and when all obstacles are convex, $K = k$. Let I_i be the number of points in object C_i equidistant to x. Therefore, $\sum_{i=1}^{K} I_i = k$.

Finally, the single object distance gradient can be written as

$$\partial d_i(x) = \mathrm{Co}\left\{\frac{x - c_1^i}{\|x - c_1^i\|}, \ldots, \frac{x - c_{I_i}^i}{\|x - c_{I_i}^i\|}\right\}. \tag{B.9}$$

The generalized gradient of the multi-object distance function can be written as

$$
\begin{aligned}
\partial D(x) &= \mathrm{Co}\{\partial d_{i_1}(x), \ldots, \partial d_{i_K}(x)\} \\
&= \mathrm{Co}\left\{\mathrm{Co}\left\{\frac{x-c_1^{i_1}}{\|x-c_1^{i_1}\|}, \ldots, \frac{x-c_{I_{i_1}}^{i_1}}{\|x-c_{I_{i_1}}^{i_1}\|}\right\}, \ldots, \mathrm{Co}\left\{\frac{x-c_1^{i_K}}{\|x-c_1^{i_K}\|}, \ldots, \frac{x-c_{I_{i_K}}^{i_K}}{\|x-c_{I_{i_K}}^{i_K}\|}\right\}\right\} \\
&= \mathrm{Co}\left\{\frac{x-c_1^{i_1}}{\|x-c_1^{i_1}\|}, \ldots, \frac{x-c_{I_{i_1}}^{i_1}}{\|x-c_{I_{i_1}}^{i_1}\|}, \ldots, \frac{x-c_1^{i_K}}{\|x-c_1^{i_K}\|}, \ldots, \frac{x-c_{I_{i_K}}^{i_K}}{\|x-c_{I_{i_K}}^{i_K}\|}\right\} \\
&= \mathrm{Co}\left\{\frac{x-c_1}{\|x-c_1\|}, \ldots, \frac{x-c_k}{\|x-c_k\|}\right\} \\
&= \mathrm{Co}\{\nabla d_1(x), \ldots, \nabla d_k(x)\}.
\end{aligned}
$$

■

B.3.3 Distance Function on a Slice

The OPP curve fragments are traces of local maxima of D, constrained to a *slice*. A slice is a set of points $\{x : \alpha(x) = \lambda\}$ where λ is a scalar and $\bigcup_{\lambda \in \mathbb{R}} \{x : \alpha(x) = \lambda\} = \mathcal{W}$. For now, we assume that a slice is a hyperplane, and that coordinates are chosen so that $\alpha(x) = x_1$. In this case, we can decompose the physical space coordinates x into "slice coordinates" y and the "sweep coordinate" λ: $x = (\lambda,\ y^T)^T$.

Definition B.14 (Single Object Distance Constrained to a Slice) The distance between a point, which is constrained to a slice, λ, and a set C_i is

$$\tilde{d}_i(y;\lambda) = d_i(x)\Bigg|_{\alpha^{-1}(\lambda)=\lambda} = \min_{q \in C_i} \|y - q\| \tag{B.10}$$

where $y \in \alpha^{-1}(\lambda)$. Hereafter, $\tilde{d}_i$ is shorthand for $\tilde{d}_i(y;\lambda)$.

See Figure B.1 for an example of the distance function plotted along a slice. At each slice point, $\tilde{d}_i$ is computed to the nearest point of the obstacle.

Typically, a robot's environment is populated with multiple obstacles, and thus we define a distance function for multiple obstacles.

Definition B.15 (Multi-object Distance Function Constrained to a Slice) The distance between a point, which is constrained to a slice, λ, and the nearest

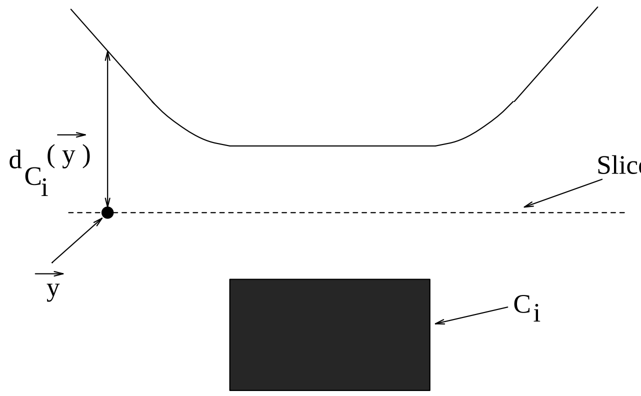

Figure B.1 Distance function plotted along a horizontal slice.

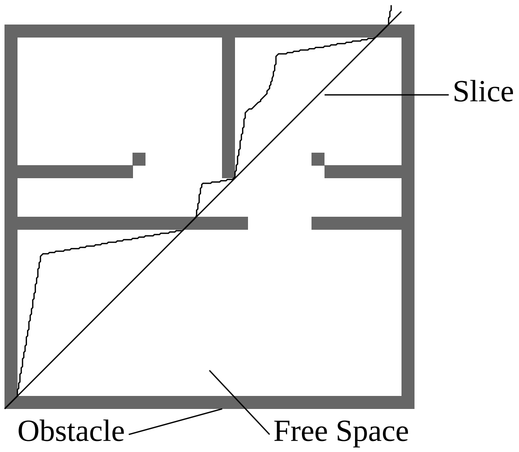

Figure B.2 Distance function plotted along a diagonal slice.

obstacle to that point is

$$\tilde{D}(y;\lambda) = \min_{i=1,\ldots,n} \tilde{d}_i(y;\lambda) \qquad y \in \alpha^{-1}(\lambda), \tag{B.11}$$

where n is the number of convex obstacles in $\mathcal{W}$.

This is the function which is maximized to generate the freeways of the OPP, and to access a GVG edge. See Figure B.2 for an example of $\tilde{D}$ plotted along a slice. However, note that D and $\tilde{D}$ are not necessarily smooth (at the local maxima), as can be seen in Figure B.2.

B.4 Extrema of Distance Function

To implement a sensor based incremental construction of a retract, we must compute the gradient of our distance measurements directly from sensor data. In particular, for the OPP freeways, the local maxima of $\tilde{D}(y;\lambda)$ needs to be determined on each slice. In order to extremize $\tilde{D}(y;\lambda)$ on a slice, we must compute its gradient with respect to the slice variables, y. In this section we show how sensor data can be used to compute this gradient, and how to reliably find and differentiate between the different required extremal points on each slice.

We want to compute the generalized gradient of $\tilde{D}(y;\lambda)$ with respect to the slice variables, y. However, our sensors give us data which can be used to construct the generalized gradient of $D(x)$ in the ambient space. This difference can be resolved as follows.

Proposition B.16 *The orthogonal projection of $\partial D(x)$ onto the y subspace is equal to the partial gradient of $D(x)$ with respect to y. In other words,*

$$\pi_y(\partial D(x)) = \partial_y D(x),$$

where π_y orthogonally projects vectors onto the y subspace and ∂_y represents partial differentiation with respect to y.

Proof: First recall that for smooth functions

$$\nabla_{x_1} f(x_1, x_2) = \pi_{x_1}\left(\nabla f(x_1, x_2)\right).$$

If there is a unique closest point, then $D(x)$ is smooth at x, and the proposition is proved. If there is not a unique closest point, by Definition 2.4,

$$\partial D(x) = \sum_{i\in I(x)} \lambda_i \nabla d_i(x) \quad \text{such that} \sum_{i\in I(x)} \lambda_i = 1 \quad \lambda_i > 0.$$

Now project this generalized gradient onto the y coordinates:

$$\pi_y(\partial D(x)) = \pi_y\left(\sum_{i\in I(x)} \lambda_i \nabla d_i(x)\right)$$
$$= \sum_{i\in I(x)} \pi_y(\lambda_i \nabla d_i(x)) = \sum_{i\in I(x)} \lambda_i \pi_y \nabla d_i(x)$$
$$= \sum_{i\in I(x)} \lambda_i \nabla_y (d_i(x)) = \partial_y(D(x)).$$

■

Thus, it is quite straightforward to compute $\partial_y \tilde{D}(y,\lambda)$ from simple distance sensor data.

An important question which must now be addressed is: How do we reliably determine if we are at a maxima, minima, or saddle point on a slice? For the local maxima, some of the local minima, and saddle points, $D(x)$ and $\tilde{D}(x)$ are not smooth. Thus, unlike the case of smooth functions, we cannot use the vanishing of $\tilde{D}$'s gradient as an indication of an extremal point. However, as the following results point out, it is possible to differentiate between the extremals. To our knowledge, these results are new to the nonsmooth analysis literature. They are equivalent to the Hessian, or curvature conditions, which classify the extremal points of smooth functions.

Proposition B.17 *Local Extrema* ([19], p38) If f attains a local minima or maxima at x, then $0 \in \partial f(x)$.

In general, the converse is not always true, but for the multi-object distance function, a converse statement follows.

Proposition B.18 *In $T_{x^*}\mathbb{R}^m$, if $0 \in \mathrm{int}(\partial D(x^*))$ then x^* is a local maximum.*

Proof: This proof relies on the following lemma whose proof appears in Section A.3.
Lemma A.20 *The origin is contained in the interior of the convex hull of a set of*

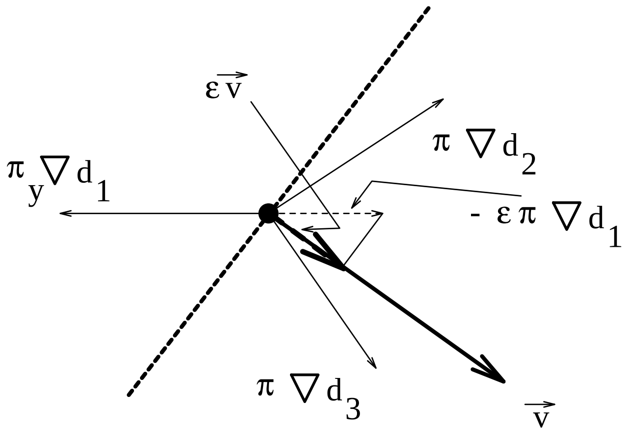

Figure B.3 Geometry of proof of Lemma A.20.

n arbitrary vectors $\{v_i \in \mathbb{R}^m : i = 1, \ldots, n\}$ if and only if there exists a v_i such that for all $w \in \mathbb{R}^m$, $\langle w, v_i \rangle > 0$.

By Lemma A.20, there exists an i for which $\{v, -\nabla d_i(x^*)\} > 0$. That is, there exists an i for which d_i decreases in the direction of v in a neighborhood of x^*. Since this is true for all v, there is always a d_i that decreases in any direction v. Therefore, for $\epsilon > 0$,

$$d_i(x + \epsilon v) < d_i(x_*) = D(x_*).$$

By definition, $D(x_* + \epsilon v) \leq d_i(x_* + \epsilon v)$, thus

$$D(x_* + \epsilon v) < D(x_*) \quad \forall \epsilon, v.$$

which implies that $D(x_*)$ is a local maxima. ■

Figure B.3 may better help visualize this proof.

Corollary B.19 Let $x^* = (\lambda, y^*)$. Let $T_{y^*}\mathbb{R}^{m-1} = \alpha^{-1}(\lambda) = \pi_y T_{x^*}\mathbb{R}^m$ be a co-dimension one slice (isometric to $\mathbb{R}^{m-1}$). We have

$$0 \in \mathrm{int}(\mathrm{Co}(\{\nabla_y \tilde{d}_i(y^*; \lambda)\})) \iff 0 \in \partial_y \tilde{D}(y^*; \lambda),$$

and if $0 \in \mathrm{int}(\mathrm{Co}(\{\nabla \tilde{d}_i(y^*; \lambda)\}))$, then y^* is a local maxima of $\tilde{D}$ on the co-dimension slice.

Proof: At $x^* = (\lambda, y^*)$, we have

$$\begin{aligned}
\partial_y \tilde{D}(y^*;\lambda) &= \partial_y D(x^*) \\
&= \pi_y \partial D(x^*) && \text{(Proposition B.16)} \\
&= \pi_y \operatorname{Co}\{\nabla d_i(x^*) : i \in I(x)\} && \text{(Proposition B.9 and Equation B.6)} \\
&= \operatorname{Co}\{\pi_y \nabla d_i(x^*) : i \in I(x)\} \\
&= \operatorname{Co}\{\nabla_y d_i(x^*) : i \in I(x)\} \\
&= \operatorname{Co}\{\nabla_y \tilde{d}_i(y^*;\lambda) : i \in I(x)\}
\end{aligned} \tag{B.12}$$

Therefore,

$$0 \in \operatorname{int}(\operatorname{Co}\{\pi_y \tilde{d}_i(y^*;\lambda)\}) \iff 0 \in \partial_y \tilde{D}(y^*;\lambda).$$

The rest of this proof parallels the proof of Proposition B.18. By Lemma A.20, there exists an i for which $\{v, -\nabla_y \tilde{d}_i(x^*)\} > 0$. That is, there exists an i for which $\tilde{d}_i$ decreases in the direction of v in a neighborhood of x^*. Since this is true for all v, there is always a $\tilde{d}_i$ that decreases in any direction v. Therefore, for $\epsilon > 0$,

$$\tilde{d}_i(y + \epsilon v) < \tilde{d}_i(y_*) = \tilde{D}(y_*).$$

By definition, $\tilde{D}(y_* + \epsilon v) \leq \tilde{d}_i(y_* + \epsilon v)$, thus

$$\tilde{D}(y_* + \epsilon v) < \tilde{D}(y_*) \quad \forall \epsilon, v$$

which implies that $\tilde{D}(y_*)$ is a local maxima. ■

The conditions for local minima and saddle points are similar, and can be proven in a similar way.

Proposition B.20 At a saddle point, D is nonsmooth, and the origin is contained in the boundary of $\partial D(x^*)$.

Proposition B.21 At a local minima of D, $0 = \partial D(x^*)$.

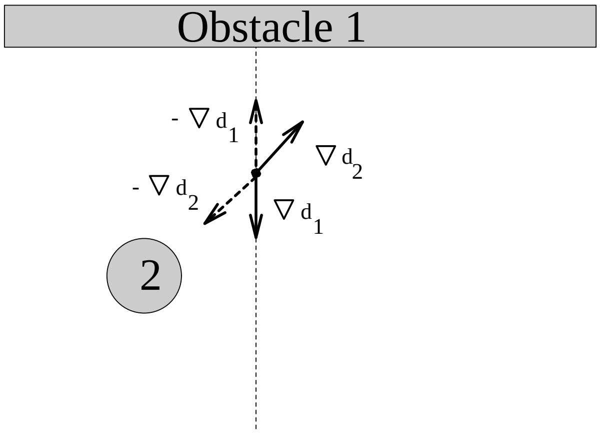

Figure B.4 Gradient of Obstacles 1 and 2 at y_*.

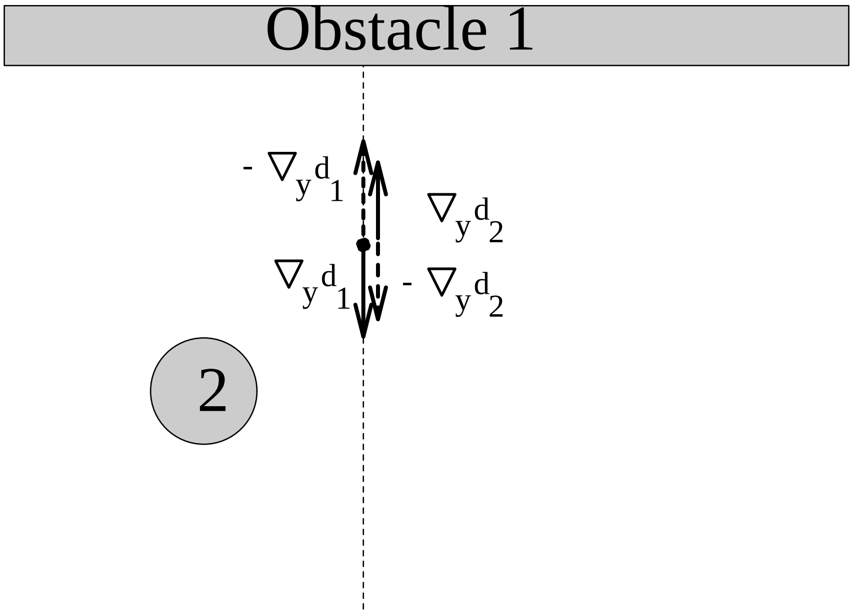

Figure B.5 Projection of the Gradients.

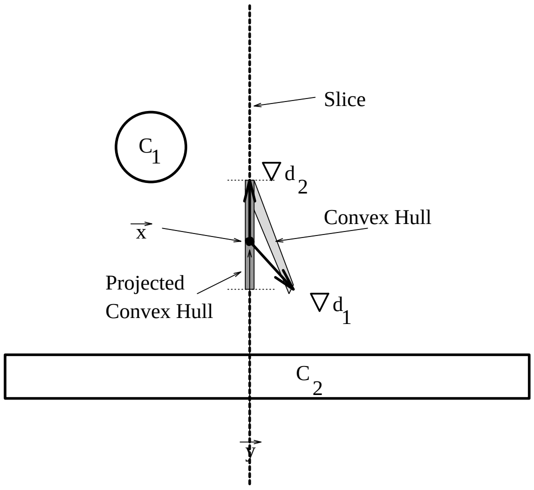

Figure B.6 Local Maximum.

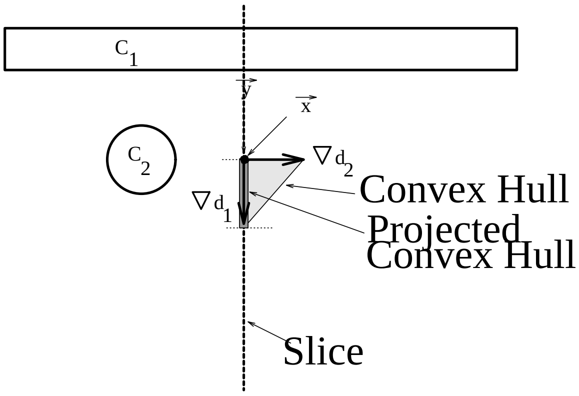

Figure B.7 Saddle Point.

Proof: At a local minima, there must be one unique closest obstacle. In this case, the generalized gradient reduces the single object distance gradient. At a local minima, this gradient vanishes. ■

Figure B.8 shows an example of a local minima. Generically, D is smooth at local minima. We term a connected local minima curve a *valley curve*.

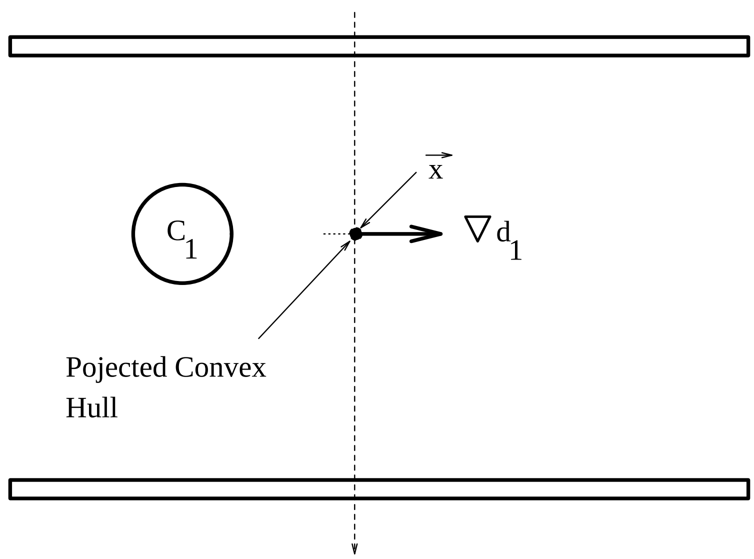

Figure B.8 Local Minimum.

Appendix C

Generalized Voronoi Complex

The *generalized Voronoi complex* is a class of structures whose properties are exploited to show connectivity of the HGVG in Chapter 3. In this chapter, after introducing some new notation, we define the generalized Voronoi complex and show it is a connected structure. As an example, we demonstrate how the generalized Voronoi diagram is a generalized Voronoi complex, and thus a connected structure.

It should be noted that the generalized Voronoi complex can be viewed as a type of dual of the GVG and HGVG. For example, when the generalized Voronoi regions form a cellular decomposition of the free space, the connectivity graph of this decomposition is a dual to the generalized Voronoi diagram. We term such a dual, a generalized Delaunay triangulation.

C.1 Basic Notation

Definition C.1 (Sub-boundary) $\dot{\partial} A$ is the subset of the boundary of a set A, such that $\dot{\partial} A$ is fully contained in A (i.e., $\dot{\partial} A = A \bigcap \partial A$).

Definition C.2 (Adjacency) Two sets A_i and A_j are *adjacent* if

$$\mathrm{cl}(A_i) \bigcap \mathrm{cl}(A_j) \neq \emptyset.$$

Definition C.3 (Sub-adjacency) *Two sets, A_i and A_j, are* sub-adjacent *if*

$$\dot{\partial} A_i \bigcap \dot{\partial} A_j \neq \emptyset.$$

C.2 Basic Definitions

Definition C.4 (Generalized Voronoi Cell) A *generalized Voronoi cell* $\mathcal{V}$ is a subset of a space X such that:

1. $\mathcal{V}$ is a connected set,
2. The sub-boundary of $\mathcal{V}$ is not empty ($\dot{\partial}\mathcal{V} \neq \emptyset$),
3. $\dot{\partial}\mathcal{V}$ is a path connected set.

An example of a cell which satisfies the above criteria is a closed simply connected set. Generalized Voronoi Regions are also examples of generalized Voronoi cells (Lemma C.10).

Lemma C.5 There exists a path from any point in a generalized Voronoi cell to its sub-boundary.

Proof: Let $\mathcal{V}$ be a generalized Voronoi cell. If $x \in \mathcal{V}$ and $y \in \mathcal{V}$, then there exists a path from x to y, because $\mathcal{V}$ is connected, by definition. If $y \in \dot{\partial}\mathcal{V}$, there still exists a path from x to y because $\dot{\partial}\mathcal{V}$ is a subset of $\mathcal{V}$. ■

Lemma C.6 If two generalized Voronoi cells, $\mathcal{V}_i$ and $\mathcal{V}_j$, are *sub-adjacent*, then their sub-boundaries are path connected (i.e., $\dot{\partial}\mathcal{V}_i \bigcup \dot{\partial}\mathcal{V}_j$ is connected).

Proof: Let $\dot{\partial}\mathcal{V}_{ij} = \dot{\partial}\mathcal{V}_i \bigcap \dot{\partial}\mathcal{V}_j$. By definition, $\dot{\partial}\mathcal{V}_i$ and $\dot{\partial}\mathcal{V}_j$ are each path connected. Since $\mathcal{V}_i$ and $\mathcal{V}_j$ are *sub-adjacent*, $\dot{\partial}\mathcal{V}_{ij}$ is not empty. Therefore, a path from any point in $\mathcal{V}_i$ to a point in $\mathcal{V}_j$ can be formed by making a path from a point in $\mathcal{V}_i$ to a point in $\mathcal{V}_{ij}$ and then by continuing this path from the point in $\mathcal{V}_{ij}$ to a point in $\mathcal{V}_j$. Therefore, $\dot{\partial}\mathcal{V}_i \bigcup \dot{\partial}\mathcal{V}_j$ is path connected. ■

Corollary C.7 If two sets $\mathcal{V}_i$ and $\mathcal{V}_j$ are each sub-adjacent to the same set $\mathcal{V}_k$, then $\dot{\partial}\mathcal{V}_i$ and $\dot{\partial}\mathcal{V}_j$ are connected.

Proof: By Lemma C.6, $\dot{\partial}\mathcal{V}_i$ is connected to $\dot{\partial}\mathcal{V}_k$ and $\dot{\partial}\mathcal{V}_j$ is connected to $\dot{\partial}\mathcal{V}_k$. By transitivity, $\dot{\partial}\mathcal{V}_i$ is connected to $\dot{\partial}\mathcal{V}_j$. ■

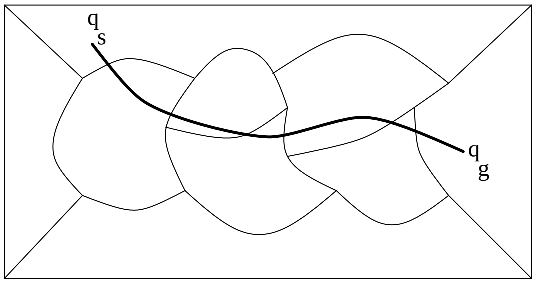

Figure C.1 Tessellation.

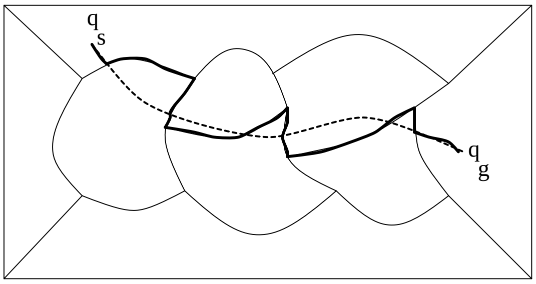

Figure C.2 Deformed Path.

C.3 Connectivity of the Generalized Voronoi Complex

Proposition C.8 The generalized Voronoi complex is connected.

Proof: The strategy of this proof is to show that a path between any two arbitrary points in the free space can be deformed into a path on the generalized Voronoi complex, $\mathcal{V}^2$.

Recall X is a connected set. Let $c\colon [0,1] \to X$ be a continuous mapping representing a continuous path between $c(0) = q_s \in X$ and $c(1) = q_g \in X$. Note, q_s and q_g are called the "start" and "goal" locations of a path, respectively. Without loss of generality, assume q_s and q_g each lie in unique cells, $\mathcal{V}_{j_s}$ and $\mathcal{V}_{j_g}$, respectively. That is, $q_s \in \mathcal{V}_{j_s}$ and $q_g \in \mathcal{V}_{j_g}$.

By Lemma C.5 there exists a path from q_s and q_g to $\dot{q_s} \in \dot{\partial}\mathcal{V}_{q_s}$ and $\dot{q_g} \in \dot{\partial}\mathcal{V}_{q_g}$, respectively.

Now, all that needs to be shown is that $\dot{q_s}$ and $\dot{q_g}$ are connected in the generalized Voronoi complex. Recall, $c\colon [0,1] \to X$ is a continuous path between q_s and q_g where $c(0) = q_s$ and $c(1) = q_g$. For each $t \in [0,1]$ assign the index of the generalized Voronoi cell for which $c(t) \in X$ belongs. In other words, let $f_c\colon [0,1] \to \{j_1, \ldots, j_k\} \subset \{j_1, \ldots, j_n\}$. Note, the image is a subset of all of the cells because the path typically does not pass through all the cells. Also note, $c(t) \in \mathcal{V}_{j_l}$ if and only if $f_c(t) = j_l$. Finally note, $f_c(0) = j_s$ and $f_c(1) = j_g$. If $c(t)$ is in multiple cells, choose any one index.

The function $f_c(t)$ prescribes a list of adjacent cells through which the path passes. Actually, in a generalized Voronoi complex, f_c prescribes a list of cells which are sub-adjacent. Consider two sub-adjacent cells, $\mathcal{V}_{j_i}$ and $\mathcal{V}_{j_{i+1}}$. By Lemma C.6, $\dot{\partial}\mathcal{V}_{j_i}$ is connected to $\dot{\partial}\mathcal{V}_{j_{i+1}}$ (through $\dot{\partial}\mathcal{V}_{j_i j_{i+1}} = \dot{\partial}\mathcal{V}_{j_i} \bigcap \dot{\partial}\mathcal{V}_{j_{i+1}}$). By re-invoking Corollary C.7, all of the sub-boundaries in the list of generalized Voronoi cells prescribed by f_c are connected. Therefore, $\dot{q}_s$ and $\dot{q}_g$ are connected. Since this is true for arbitrary q_s and q_g, the generalized Voronoi complex is connected.

In summary, in a connected space, X, a path between q_s and q_g can be formed with three connected components: (1) between q_s and $\dot{q}_s$, (2) one along the generalized Voronoi complex between $\dot{q}_s$ and $\dot{q}_g$, and (3) one between $\dot{q}_g$ and q_g. Since this is true for arbitrary q_s and q_g in a bounded space, the generalized Voronoi complex is connected. ■

C.4 Generalized Voronoi Diagram

Proposition C.9 (The generalized Voronoi diagram is Connected) In an m-dimensional space, the generalized Voronoi diagram, $\mathfrak{F}^2$, is connected.

Proof: The union of all two-equidistant faces is the generalized Voronoi diagram. By Proposition 2.11, two-equidistant faces are on the boundary of the generalized Voronoi region. Once it is established that a generalized Voronoi region is a type of generalized Voronoi cell and the generalized Voronoi regions form a generalized Voronoi complex, when $X = \mathcal{FS}$, then this proof is a simple consequence of Proposition C.8.

Lemma C.10 Generalized Voronoi regions are generalized Voronoi cells.

Proof: By Definition C.4, a generalized Voronoi region, $\mathcal{F}_i$, is a generalized Voronoi cell if it satisfies the following properties:

1. $\mathcal{F}_i$ is a connected set;

2. $\exists\, \dot{\partial}\mathcal{F}_i \neq \emptyset$; and

3. $\dot{\partial}\mathcal{F}_i$ is a connected set.

By the generalized star shape property of generalized Voronoi regions, the generalized Voronoi regions are connected.

By the boundedness assumption, there exists at least one other obstacle, C_j, adjacent to C_i such that there exists a two-equidistant face, $\mathcal{F}_{ij}$, between C_i and C_j. By definition of the two-equidistant face (Definition 2.5), $\mathcal{F}_{ij} \subset \mathcal{F}_i$. Furthermore, by Proposition 2.11, $\mathcal{F}_{ij} \subset \partial\mathcal{F}_i$. Therefore, the generalized Voronoi region $\mathcal{F}_i$ has a non-empty sub-boundary. In fact, $\dot{\partial}\mathcal{F}_i = \bigcup_j \mathcal{F}_{ij}$.

Ó'Dúnlaing and Yap [34] define a continuous function, $Im(x)$, which maps points from the generalized Voronoi region onto the generalized Voronoi diagram. Since this function is continuous, then the boundary of the generalized Voronoi region is a connected set because it is the image of the generalized Voronoi region, which is a connected set. ▼

■

Bibliography

Bibliography

[1] R. Abraham, J.E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications.* Springer-Verlag, New York, 2 edition, 1988.

[2] F. Aurenhammer. Voronoi Diagrams — A Survey of a Fundamental Geometric Structure. *ACM Computing Surveys*, 23:345–405, 1991.

[3] D. Avis and B.K. Bhattacharya. Algorithms for Computing d-dimensional Voronoi Diagrams and Their Duals. *Advances in Computing Research*, 1:159–180, 1983.

[4] J. Borenstein and J. Koren. Real-time Onstacle Avoidance for Fast Mobile Robots in Cluttered Environments. In *IEEE Conference of Robotics and Automation*, pages 572–577, Cincinnati, Ohio, May 1990.

[5] R.A. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal on Robotics and Automation*, RA-2, March 1986.

[6] J.F. Canny. *The Complexity of Robot Motion Planning.* MIT Press, Cambridge, MA, 1988.

[7] J.F. Canny and B. Donald. Simplified Voronoi Diagrams. *Discrete Comput. Geometry*, pages 219–236, 1988.

[8] J.F. Canny and M.C. Lin. An Opportunistic Global Path Planner. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1554–1559, Cincinnati, Ohio, 1990.

[9] J.F. Canny and M.C. Lin. An Opportunistic Global Path Planner. *Algorithmica*, 10:102–120, 1993.

[10] B. Chazelle. Convex partition of polyhedra: A lower bound and worst-case optimal algorithm. *SIAM Journal on Computing*, 13(3):488–507, 1984.

[11] G.S. Chirikjian. *Theory and Applications of Hyper-Redundant Robotic Manipulators.* PhD thesis, California Institute of Technology, Pasadena, CA, 1992.

[12] G.S. Chirikjian and J.W. Burdick. A Modal Approach to Hyper-Redundant Manipulator Kinematics. *IEEE Trans. on Robotics and Automation*, June 1994.

[13] G.S. Chirikjian and J.W. Burdick. Kinematically Optimal Hyper-Redundant Manipulator Configurations. *IEEE Trans. on Automation and Robotics*, December 1995.

[14] G.S. Chirikjian and J.W. Burdick. The Kinematics of Hyper-Redundant Robotic Locomotion. *IEEE Trans. on Automation and Robotics*, December 1995. Submitted.

[15] H. Choset and J.W. Burdick. Sensor Based Planning and Nonsmooth Analysis. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3034–3041, San Diego, CA, 1994.

[16] H. Choset and J.W. Burdick. Sensor Based Planning, Part I: The Generalized Voronoi Graph. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, 1995.

[17] H. Choset and J.W. Burdick. Sensor Based Planning, Part II: Incremental Construction of the Generalized Voronoi Graph. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, 1995.

[18] H. Choset and J.W. Burdick. Sensor Based Planning for a Planar Rod Robot. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, 1996.

[19] F. H. Clarke. *Optimization and Nonsmooth Analysis.* Society of Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[20] E. Gat. ALFA: A Language for Programming Reactive Robotic Control Systems. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1116–1121, Sacramento, CA, April 1991.

[21] E. Gat. Integrating Reaction and Planning in Heterogenous Asynchronous Architecture for Controlling Real World Mobile Robots. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 1992.

[22] E. Gat and G. Dorais. Robot Navigation by Conditional Sequencing. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1293–1299, San Diego, CA, May 1994.

[23] A. J. Golman and A. W. Tucker. *Linear Inequalities and Related Systems*, chapter Polyhedral Convex Cones. Princeton University Press, Princeton, 1956.

[24] H.B. Keller. *Lectures on Numerical Methods in Bifurcation Problems.* Tata Institute of Fundamental Research, Bombay, India, 1987.

[25] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5:90–98, 1896.

[26] V. Klee. On the Complexity of D-dimensional Voronoi Diagrams. *Arch. Math.*, 34:75–80, 1980.

[27] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, July 1994.

[28] J.C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991.

[29] M.C. Lin and J.F. Canny. Efficient algorithms for incremental distance computation. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 1991.

[30] T. Lozano-Perez and M.A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, 1979.

[31] V. Lumelsky and A. Stepanov. Path Planning Strategies for Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape. *Algorithmica*, 2:403–430, 1987.

[32] J.E. Marsden. *Elementary Classical Analysis.* W. H. Freeman and Company, New York, 1974.

[33] V-D. Nguyen. Constructing force-closure grasps. *IJRR*, 7(3):3–16, June 1988.

[34] C. Ó'Dúnlaing and C.K. Yap. A "Retraction" Method for Planning the Motion of a Disc. *Algorithmica*, 6:104–111, 1985.

[35] N.S.V. Rao, S. Kareti, W. Shi, and S.S. Iyenagar. Robot Navigation in Unknown Terrains: Introductory Survey of Non-Heuristic Algorithms. *Oak Ridge National Laboratory Technical Report*, ORNL/TM-12410:1–58, July 1993.

[36] N.S.V. Rao, N. Stolzfus, and S.S. Iyengar. A Retraction Method for Learned Navigation in Unknown Terrains for a Circular Robot. *IEEE Transactions on Robotics and Automation*, 7:699–707, October 1991.

[37] D. Reznik and V. Lumelsky. Motion Planning with Uncertainty for Highly Redundant Kinematic Structures I. 'Free Snake' Motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, N.C., 1992.

[38] E. Rimon and J.F. Canny. Construction of C-space Roadmaps Using Local Sensory Data — What Should the Sensors Look For? In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 117–124, San Diego, CA, 1994.

[39] E. Rimon and D. E. Koditschek. Exact Robot Navigation Using Artificial Potential Functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, October 1992.

[40] P.F. Rowat. "Representing the Spatial Experience and Solving Spatial Problems in a Simulated Robot Environment". In *PhD. Thesis*, University of British Columbia, 1979.

[41] S. Scheimberg and P.R. Oliveira. Descent algorithm for a class of convex nondifferentiable functions. *Journal of Optimization Theory and Applications*, 72:269–297, February 1992.

[42] J.T. Schwartz and C.K. Yap, editors. *Advances in Robotics: Algorithmic and Geometric Apsects of Robotics*, volume 1. Lawrence Erlbaum Associates, New Jersey, 1987.

[43] C.J. Taylor and D.J. Kriegman. Vision-Basied Motion Planning and Exploration Algorithms for Mobile Robots. In *Proc. of Workshop on the Algorithmic Foundations of Robotics*, San Francisco, CA, February 1994.