Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph

Howie Choset, *Carnegie Mellon University, Pittsburgh, PA, USA*
Joel Burdick, *California Institute of Technology, Pasadena, CA, USA*

Abstract *The* hierarchical generalized Voronoi graph *(HGVG) is a* roadmap *that can serve as a basis for* sensor based robot motion planning. *A key feature of the HGVG is its incremental construction procedure that uses only* line of sight *distance information. This work describes basic properties of the HGVG and the procedure for its incremental construction using local range sensors. Simulations and experiments verify this approach.*

1 Introduction

Sensor based motion 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's geometry prior to planning. Sensor based planning is important for realistic deployment of robots because: (1) the robot often has no a priori knowledge of the world; (2) the robot may have only a coarse knowledge of the world because of limited computer memory; (3) the world model is bound to contain inaccuracies which can be overcome with sensor based planning strategies; and (4) the world is subject to unexpected occurrences or rapidly changing situations.

This work addresses two primary problems in sensor based motion planning for static environments. The first problem deals with the case of when a robot is given a target location and it always knows its current location (i.e., it has an on-board dead reckoning system). Assuming no a priori knowledge about its environment, the robot must find a collision-free path to the goal, based on sensory measurement. In the second problem, the robot is placed in a bounded environment with no a priori knowledge about that environment. Using only its on-board sensors and a dead reckoning system, the robot must build a complete *roadmap* of the bounded environment. A roadmap is a collection of one-dimensional curves that capture the important topological and geometric properties of the robot's environment. Roadmaps have the following properties: *accessibility*, *departability*, and *connectivity*. These properties imply that 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). The solution to this second problem automatically supplies a solution to the first problem, and so we will focus our attention on the second problem. The *hierarchical generalized Voronoi graph* (HGVG), defined in this work, is a roadmap which can be incrementally constructed using line of sight sensor data.

2 Relation to Previous Work

Many sensor based planners are heuristic and work well under a variety of conditions. Nevertheless, there are no proofs of correctness that guarantee a path can be found and furthermore, there do not exist well established thresholds for when heuristic algorithms fail. 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 [2]. A hierarchy of cooperating behaviors forms more complicated behaviors such as exploration. An extension of this type of approach is called sequencing [11]. Since there are strong experimental results indicating the utility of these approaches, some of these algorithms may provide a future basis for provably correct sensor based planners.

There are many non-heuristic algorithms for which provably correct solutions exist in the plane (see [15] for an overview). For example, Lumelsky's "bug" algorithm [13] is one of the first provably correct sensor based schemes to work in the plane. However, this algorithm (like many described in [15]) requires knowledge of the goal's location during the planning process. Hence, the robot can't search for a goal "beacon." 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.

One approach to sensor based motion planning is to adapt the structure of a provably correct, or "complete," classical motion planning scheme to a sensor based implementation. Roadmaps are one of the complete classical methods. An example of a complete roadmap scheme is Canny and Lin's Opportunistic Path Planner (OPP) [4]. Rimon adapted this motion planning scheme for sensor based use [17], but, connectivity of the roadmap in [17] cannot be guaranteed without "active perception." Furthermore, from a practical point of view, there are two detractions to Rimon's method. First, to construct the roadmap, the robot must contain "interesting critical point" and "interesting saddle point" sensors, whose implementation is not well described. Second, a robust, detailed, and efficient procedure for constructing the roadmap edges from sensor data is not presented.

The HGVG 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 is the set of points equidistant to two or more points (sometimes termed sites) in the plane.) The GVD was first used for motion planning in [18]. Active research in applying the GVD to motion planning began with Ó'Dúnlaing and Yap [14], who considered motion planning for a disk in the plane. However, the method in [14] 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 [16], 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., [1]) describes the *Voronoi graph*, an extension of the Voronoi diagram 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 Section 5, 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 [3].

The GVG can be thought of as the natural extension of the GVD into higher dimensions. However, unlike the GVD, the GVG is not necessarily connected in dimensions greater than two, and thus, in general, is not a roadmap. Therefore, 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).

3 Contributions

Although the HGVG is applicable to classical motion planning, the intended use of the HGVG is for sensor based planning. When full knowledge of the world is available to the robot, we make no claim that the HGVG has any clear advantage over other one-dimensional roadmaps used in multi-dimensional configuration spaces, though our experience shows that the construction of this roadmap is efficient. However, since the HGVG is defined in terms of distance information, it lends itself to sensor based construction. We describe an incremental construction technique for the HGVG. 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

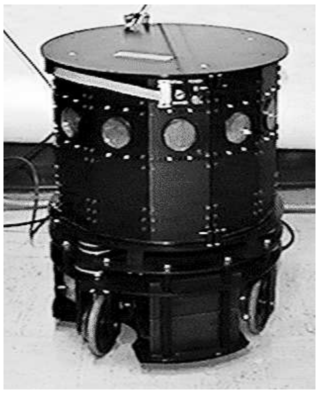

Figure 1: *Mobile robot with sonar ring.*

require any artificial landmarks, complicated obstacle segmentation, nor abstract sensors. Additionally, this incremental construction technique can be applied to construct the edges of other roadmaps, such as Canny and Lin's Opportunistic Path Planner, Rimon's extension of the OPP, and wall following algorithms.

4 Distance Function

The HGVG and its properties are based on the following workspace distance function definitions. Assume that the robot is a point operating in a work space, $\mathcal{W}$, which is a subset of an m-dimensional Euclidean space. $\mathcal{W}$ is populated by convex obstacles $C_1, \ldots, C_n$. 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 $\{C_i\}$. The free space, $\mathcal{FS}$, is the subset of $\mathcal{W}$ not occupied by obstacles.

The distance between a point x and a convex set C_i is termed the single object distance function and is defined as

$$d_i^X(x) = \min_{c_0 \in C_i} \|x - c_0\|, \tag{1}$$

where $\|\cdot\|$ is the two-norm in $\mathbb{R}^m$. The gradient of $d_i^X(x)$ is

$$\nabla d_i^X(x) = \frac{x - c_0}{\|x - c_0\|}. \tag{2}$$

The vector $\nabla d_i^X(x)$ is a unit vector in the direction from c_0 to x, where c_0 is the nearest point to x in C_i.

To compute Equations (1) and (2) from sensor data, one need only know the distance and direction to the nearest point on C_i. For convex sets, the closest point is always unique.

Typically, the environment is populated with multiple obstacles, and thus we define a multi-object distance function as $D^X(x) = \min_i d_i^X(x)$.

The single object distance function measures the distance to obstacles that may not be within line of sight of the robot. We term this distance function the "X-ray" distance function because it assumes the robot has X-ray vision.

Alas, for the purposes of sensor based motion planning, the robot can only measure distances to obstacles it can see. To this end, we define the *single object visible distance function* which measures distance to nearby obstacles that are *within visible-line of sight.* A point c is *within line of sight* of x if there exists a straight line segment that connects x and c without penetrating any obstacle. That is, c is within line of sight of x if for all $t \in [0,1]$, $(x(1-t)+ct)$ is a point 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.,

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

Let $c_i = \text{argmin}\, d_i^X(x)$, i.e., c_i is the closest point on the convex obstacle C_i to x. The *single object visible distance function* is the distance to obstacle C_i if c_i is within line of sight of x. In this case, C_i is within *visible*-line of sight of x. If C_i is not within visible-line of sight of x, then the distance to C_i is infinite. We denote the visible distance function as $d_i(x)$. If the visible distance function has a finite value at a point x, then its gradient is defined by Equation (2) and is denoted $\nabla d_i(x)$. The visible multi-object distance function is

$$D(x) = \min_i d_i(x). \tag{3}$$

In this work, we use the visible distance functions. That is, our definition of the GVG and HGVG in terms of the visible distance function is unique.

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 1). The sensor measurement provides an approximate value of the distance function. The centerline of the sensor's measurement axis (which is orthogonal to the sensor face) approximates the distance gradient. Stereo vision, depth from focus, or laser range fingers can also be used to provide depth and gradients to points on surfaces in the surrounding environment. With this distance function, which makes use of line of sight information, we can define the HGVG, and related structures such as the generalized Voronoi graph.

5 The Generalized Voronoi Graph

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*,

$$\mathcal{S}_{ij} = \{x \in \mathcal{W} \backslash (C_i \bigcup C_j) : d_i(x) - d_j(x) = 0\}. \tag{4}$$

See Figure 2. Of particular interest is the subset of $\mathcal{S}_{ij}$ termed the *two-equidistant surjective surface*,

$$\mathcal{SS}_{ij} = \mathrm{cl}\{x \in \mathcal{S}_{ij} : \nabla d_i(x) \neq \nabla d_j(x)\}. \tag{5}$$

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. This definition is required to deal with non-convex sets that are defined as the finite union of convex sets. See Figure 3. If $\mathcal{W}$ is solely populated with disjoint convex obstacles, then $\mathcal{SS}_{ij} = \mathcal{S}_{ij}$ for all i, j.

Using the Pre-image Theorem, it can be shown that $\mathcal{SS}_{ij}$ is an $(m-1)$-dimensional manifold in $\mathbb{R}^m$ (i.e., $\mathcal{SS}_{ij}$ has co-dimension one) [6].

Definition 1 (Two-Equidistant Face) The *two-equidistant face*,

$$\mathcal{F}_{ij} = \mathrm{cl}\{x \in \mathcal{SS}_{ij} : d_i(x) = d_j(x) \leq d_h(x) \quad \forall h \neq i, j\}, \tag{6}$$

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

See Figure 4. In keeping with the conventions of the Voronoi diagram literature, a two-equidistant face is also termed a *generalized Voronoi face*.

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

$$\mathfrak{F}^2 = \bigcup_{i=1}^{n-1} \bigcup_{j=i+1}^{n} \mathcal{F}_{ij} \tag{7}$$

Since $\mathfrak{F}^2$ is the set of points equidistant to the two or

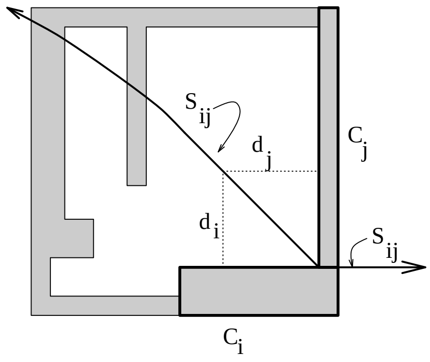

Figure 2: *The solid line represents $\mathcal{S}_{ij}$, the set of points equidistant to obstacles C_i and C_j. The dotted lines emphasize that at $x \in \mathcal{S}_{ij}$, $d_i(x) = d_j(x)$.*

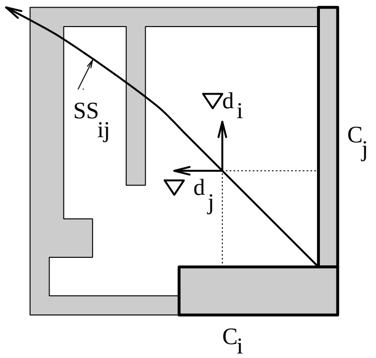

Figure 3: *The solid line represents $\mathcal{SS}_{ij}$, the set of points equidistant to obstacles C_i and C_j such that the two closest points are distinct. Note, it is also unbounded and only has one component.*

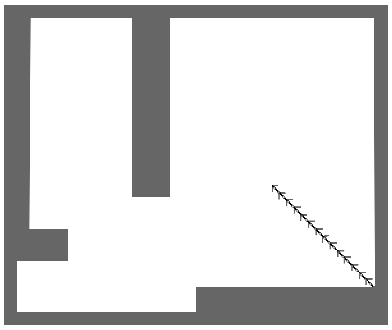

Figure 4: *The ticked solid line is the set of points equidistant and closest to obstacles C_i and C_j.*

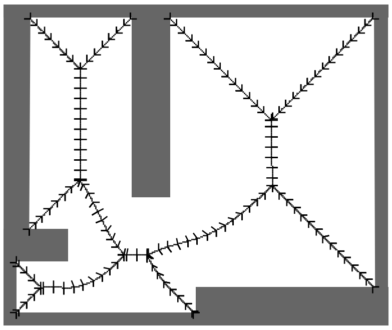

Figure 5: *The ticked solid lines is the set of points equidistant to two obstacles, such that each edge fragment is closest to the equidistant obstacles.*

more closest points on the boundary of $\mathcal{W}$, it is the generalized Voronoi diagram (GVD) of $\mathcal{W}$. See Figure 5. Since it can be shown that $\mathcal{F}_{ij}$ is an $(m-1)$-dimensional manifold with an $(m-2)$-dimensional boundary, $\mathfrak{F}^2$ has co-dimension one, but is not necessarily a manifold.

To define the GVG, we continue to define lower dimensional subsets of $\mathcal{W}$. The *k-equidistant face*, $\mathcal{F}_{i_1 i_2 \ldots i_k}$, is the $(m-k+1)$-dimensional set of points that are equidistant to objects $C_{i_1}, C_{i_2}, \ldots, C_{i_k}$ such that each point is closer to objects $C_{i_1}, C_{i_2}, \ldots, C_{i_k}$ than to any other object and no two distance gradient vectors are collinear. The *k-Voronoi set*, $\mathfrak{F}^k$, is simply the union of all $(m-k+1)$-dimensional k-equidistant faces.

$$\begin{aligned} \mathcal{F}_{i_1 i_2 \ldots i_k} &= \mathcal{F}_{i_1 i_2} \bigcap \cdots \bigcap \mathcal{F}_{i_1 i_k}, \\ &= \mathcal{F}_{i_1 i_2 \ldots i_{k-1}} \bigcap \mathcal{F}_{i_1 i_k}, \\ \mathfrak{F}^k &= \bigcup_{i_1=1}^{n-k+1} \bigcup_{i_2=i_1+1}^{n-k+2} \cdots \bigcup_{i_k=i_{k-1}+1}^{n} \mathcal{F}_{i_1 i_2 \ldots i_k} \end{aligned} \tag{8}$$

Generalized Voronoi Graph Definition. In m dimensions, a *generalized Voronoi edge* and a *generalized Voronoi vertex* are respectively an m-equidistant face, $\mathcal{F}_{i_1 \ldots i_m}$, and an $m+1$-equidistant face, $\mathcal{F}_{i_1 \ldots i_{m+1}}$. Generically (i.e, when equidistant faces transversally intersect), the Pre-image Theorem asserts that generalized Voronoi edges are one-dimensional and that the generalized Voronoi vertices are points. Since generalized Voronoi edges meet at generalized Voronoi vertices, generalized Voronoi vertices are termed *meet points*. Using these definitions, we can define:

Definition 2 (Generalized Voronoi Graph) *The generalized Voronoi graph* (GVG) is the collection of generalized Voronoi edges and meet points. That is,

$$GVG = (\mathfrak{F}^m, \mathfrak{F}^{m+1}). \tag{9}$$

The GVG's edges comprise the set of points equidistant to m objects, such that each point is closer to m objects than to any other object. *An important characteristic of the GVG is that it is defined in terms of a visible distance function, which can be computed readily from sensor data.* It is this feature that makes the GVG useful for sensor based motion planning. Our definition of the Voronoi graphs and diagrams in terms of a *visible* distance function is unique.

Example 1 *Figure 6 depicts a generalized Voronoi graph for a rectangular enclosure in $\mathbb{R}^3$. The GVG edges, delineated by solid lines, constitute the locus points equidistant to three obstacles, and the meet points are where the GVG edges intersect.* ♦

6 Incremental Construction of the GVG

A key feature of the GVG is that it can be incrementally constructed using line of sight range information.

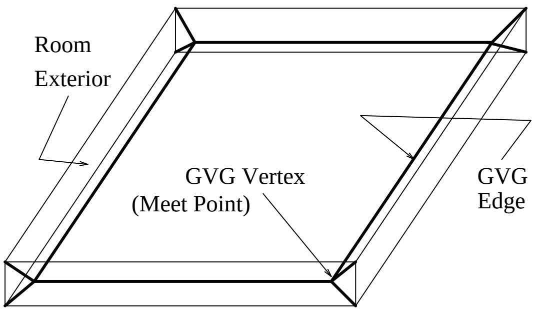

Figure 6: *The generalized Voronoi graph in a rectangular enclosure. The solid lines represent the GVG edges, which meet at GVG vertices, or "meet points."*

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 such a single vantage point.

The incremental construction techniques described in this section provide a rigorous approach to constructing the GVG using only line of sight sensory information. It is worth noting that these incremental construction procedures can be the basis of a numerical method to construct a roadmap when full geometry of the world is available. Furthermore, we have found that our incremental construction procedures are not only sensor implementable, but numerically efficient. Finally, the incremental construction techniques described in this section can be adapted to other sensor based planning methods such as the OPP (described in [4, 17]).

The HGVG's properties of accessibility, departability and connectivity translate to *incremental accessibility, incremental departability*, and *traceability,* respectively, in the incremental construction procedure. This section describes how to move onto (incremental accessibility) and trace along (traceability) the GVG *using only local information.* Incremental departability is described in Section 8. The algorithm is verified by simulations and experiments that are reviewed in Section 7.

6.1 Incremental Accessibility

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 [5], it was 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* [10]. The generalized gradient of D was shown in [5] to be

$$\partial D(x) = \mathrm{Co}\{\nabla d_i(x) : \; \forall i \in I(x)\}, \tag{10}$$

where $I(x)$ is the set of indices where $d_i(x) = D(x)$, and where Co denotes convex hull.

Furthermore, it was shown in [5] 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. It is worth noting that this local maxima is determined solely from first order information. Using this result and the following two lemmas, we can conclude that if x is a local maxima of D, then it is equidistant to $m+1$ obstacles.

Lemma 1 *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 2 (Goldman and Tucker) *It requires a minimum of $(m+1)$ vectors to positively span $\mathbb{R}^m$.*

6.2 Traceability

In an incremental context, the property of connectivity is interpreted as *traceability*. More specifically, traceability implies that using only local data, the robot can: (1) "trace" the GVG (or HGVG) edges; (2) determine all of the edges that emanate from a meet point; (3) change directions at a meet point, and thereby begin tracing new edges; and (4) determine when to terminate the tracing procedure.

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

[4] method and its sensor based adaptation [17] 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 [5].

Our approach to edge construction borrows ideas from numerical continuation methods [12]. Continuation methods trace the roots of the expression $G(y, \lambda) = 0$ 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 7). At x, let the hyperplane spanned by coordinates $z_2, \ldots, z_m$ be termed the "normal slice plane." We can thus decompose the local coordinates into $x = (y, \lambda)$, where $\lambda = z_1$ is termed the local "sweep" coordinate and $y = (z_2, \ldots, z_m)$ are the "slice" coordinates. Now define the function $G\colon \mathbb{R}^{m-1} \times \mathbb{R} \to \mathbb{R}^{m-1}$ as follows:

$$G(y, \lambda) = \begin{bmatrix} (d_1 - d_2)(y, \lambda) \\ (d_1 - d_3)(y, \lambda) \\ \vdots \\ (d_1 - d_m)(y, \lambda) \end{bmatrix} \tag{11}$$

The function $G(y, \lambda)$ assumes a zero value only on the GVG. Hence, if the Jacobian of G, $\nabla_y G$, is surjective, then the implicit function theorem implies that the roots of $G(y, \lambda)$ locally define a generalized Voronoi edge as λ is varied.

By numerically tracing the roots of G, we can locally construct an edge. While there are a number of such techniques [12], we use an adaptation of a common predictor-corrector scheme. 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). This tangent is computed using line of sight information as follows:

Proposition 1 *The tangent to a GVG edge at x is defined by the vector orthogonal to the hyperplane that contains the m closest points: $c_1, \ldots, c_m$ of the m closest objects, $C_1, \ldots, C_m$.*

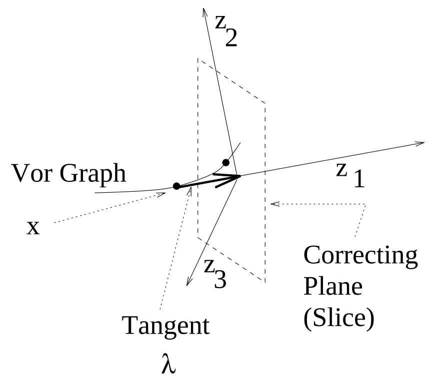

Figure 7: *Sketch of Continuation Method*

The proof of the above appears in [7].

In general, this "prediction" step will take 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" (Figure 7), which is a plane parallel to the normal slice at distance $\Delta\lambda$. The correction step finds the location where the GVG intersects the correcting plane. (Figure 7)

The explicit correction procedure is an iterative Newton's Method. If y^k and λ^k are the kth estimates of y and λ, the $k+1$st iteration is defined as

$$y^{k+1} = y^k - \left(\nabla_y G\right)^{-1} G(y^k, \lambda^k) \tag{12}$$

where $\nabla_y G$ is evaluated at (y^k, λ^k). The following proposition, whose proof appears in [7], guarantees that Equation (12) is well defined:

Proposition 2 *(Equidistant Surface Full Rank Property) The matrix $\nabla_y G(y, \lambda)$ has full rank (i.e., has rank $(m-1)$) in a neighborhood of the GVG on the correcting plane.*

Practically speaking, this result states that the numerical procedure defined by Equation (12) will be robust for reasonable errors in robot position, sensor errors, and numerical round off.

There are several things worth noting about this method. First, to evaluate $G(y, \lambda)$ and $\nabla_y G(y, \lambda)$, 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, $\nabla_y G(y, \lambda)$ 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).

6.3 Terminating Conditions

So far, we have shown that the robot can access and trace a GVG edge. Due to the boundedness of the robot's environment, GVG edges must terminate. Most GVG edges have meet points for end points, but the some GVG edges may have other types of end points. A GVG edge may terminate on a *boundary point*, denoted $C_{i_1 \ldots i_m}$, where m obstacles intersect. A GVG edge may also terminate at a *floating boundary point*, denoted $FC_{i_1 \ldots i_m}$. At a floating boundary point, two distance function gradient vectors become collinear. The following proposition, whose proof appears in [9], guarantees that the only terminating conditions for a GVG edge are meet points, boundary points, and floating boundary points.

Proposition 3 *Given that equidistant faces transversally intersect 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.*

This result is apparent by inspecting the definition of a GVG edge in $\mathbb{R}^3$:

$$\begin{aligned}\mathcal{F}_{ijk} = \mathrm{cl}\{x \in \mathcal{FS} : \nabla d_i(x) \neq \nabla d_j(x) \text{ and} \\ 0 < d_i(x) = d_j(x) = d_k(x) < d_h(x)\} \quad (13)\end{aligned}$$

The boundary of the GVG edge associated with the inequality $\nabla d_i(x) \neq \nabla d_j(x)$ is a floating boundary point (where two gradient vectors become collinear); the boundary of the GVG edge associated with the first inequality $0 < d_i(x)$ is a boundary point (where distance to the environment is zero); and, the boundary associated with the last inequality is a meet point.

Incremental construction of the GVG is akin to a graph search where GVG edges are the "edges" and the meet points, boundary points, and floating 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 off 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 off that direction as being explored. If the robot reaches another unvisited meet point, the above procedure is recursively repeated. When the robot hits a boundary point or a floating 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 fragment when there are no more unexplored directions associated with any meet point. If the robot is looking for a particular destination whose coordinates are known, then the robot can invoke graph searching techniques, such as the A-star algorithm, to control the tracing procedure.

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 because of sensor error. Furthermore, since the robot is taking finite sized steps while tracing an edge, it is unlikely that the robot will pass exactly through an $(m+1)$-equidistant point. However, as shown in Figure 8, 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.

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. 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

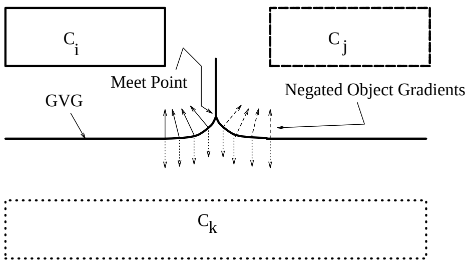

Figure 8: *Meet Point Detection*

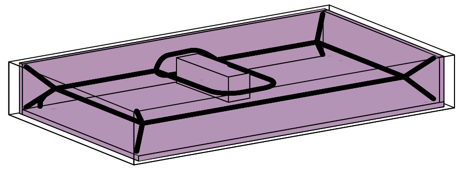

Figure 9: *GVG of a 3-dimensional box with a long box which is located off-center in the interior.*

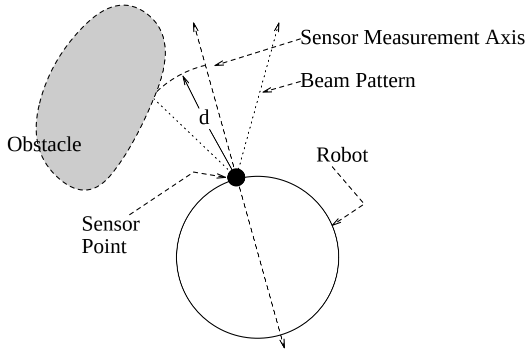

Figure 10: *Simplified distance measurement sensor model.*

$C_1, \ldots, C_m$. Proposition 1 yields the one-dimensional tangent space to the generalized Voronoi edge corresponding to these m objects. If v is a tangent vector computed from Proposition 1, the robot must determine if it should depart the meet point in the $+v$ or $-v$ direction. Let $d_1(x) = d_2(x) = \cdots = d_m(x) = d_{m+1}(x)$ be the distances to the $m+1$ closest objects at a meet point. 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 $-v$. Recall that a generalized Voronoi edge is closer to the m objects that define it, than any other object. This effects motion away from C_{m+1}.

So far, we have defined the GVG and demonstrated some of its properties. In particular, we showed that using line of sight information, a robot can access the GVG and incrementally trace out its edges. These properties are verified in the following section. A discussion of the properties of connectivity and departability are deferred to Section 8.

7 Simulations and Experiments

Planar Simulations. A planar simulator has validated this approach for a point or circularly symmetric robot operating in the plane. Figure 4 contains an example of a bounded environment in which our algorithm was tested. In this figure, the robot has accessed the GVG and traced one GVG edge. The ticked solid lines represent the planar GVG (also the GVD); these are the locus of points equidistant to the two nearest obstacles. The ticks point to the nearest obstacles. Figure 5 shows the final simulation result.

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 this simulator was written by Brian Mirtich at Berkeley. Currently, the linking procedures (described in the next section) are under development. See Figure 9 for final results of GVG tracing.

Experimental Results. Another advantage of this approach is that the HGVG can be incrementally constructed from raw sensor data. Experiments were performed on a mobile robot with a ring of ultrasonic range sensors, depicted in Figure 1. We assume the sensors measure distance to nearby obstacles, along a fixed direction termed the *sensor measurement axis.* The sensor measurement axis is a function of the robot's position and orientation (See Figure 10). It is shown in [9] that the distances to individual obstacles correspond to local minima in the sensor array. The distance gradient is the unit vector pointing along the sensor measurement axis. An example is depicted in

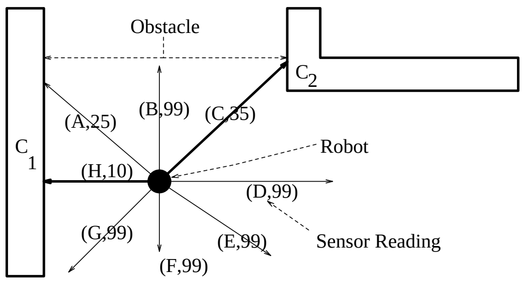

Figure 11: *Sensors H and Sensors C are associated with the distances to the two closest obstacles.*

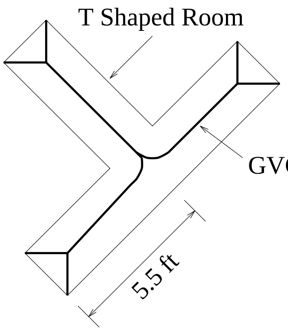

Figure 12: *T-shaped Room with Actual GVG.*

Figure 11 where a robot with eight sensors and their measurements is drawn.

The result of one experiment in a "T-shaped" room using the mobile robot is shown in Figure 12 (theoretical GVG) and in Figure 13 (experimental GVG). The small in Figure 13 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 localization.

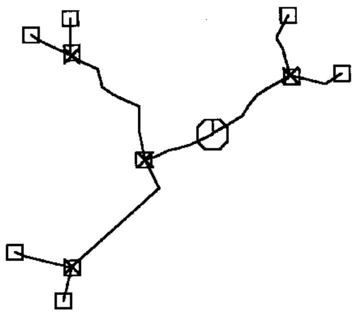

Figure 13: *Experimental GVG.*

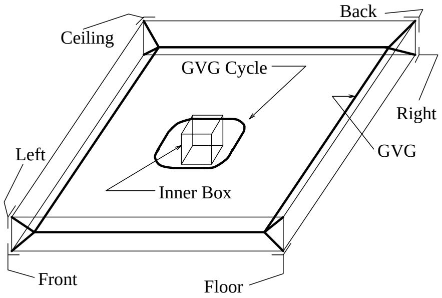

Figure 14: *Disconnected GVG.*

8 Hierarchical Generalized Voronoi Graph

Although the planar GVG and the particular GVG in Figure 9 are connected, the GVG is not guaranteed to be connected in work space dimensions greater than two. Figure 14 contains an example of a disconnected GVG with two connected components: (1) an outer GVG network similar to the one described in Example 1 and (2) an inner GVG network which forms a halo-like structure around the inner box. This section outlines our complete roadmap, which is the GVG augmented by additional structures that are 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. In other words, we recursively invoke the GVG construction procedure on lower dimensional faces. For example, *a second order generalized Voronoi graph*, de-

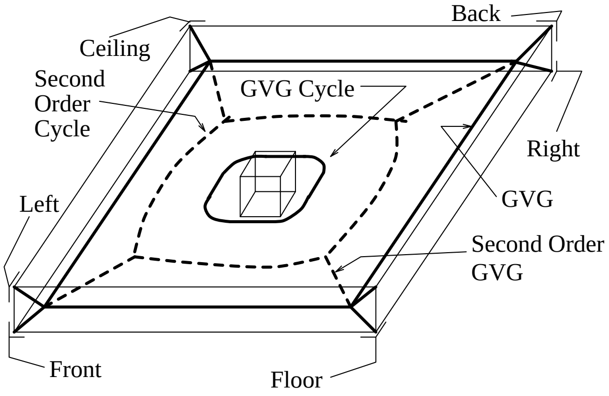

Figure 15: *The GVG^2 is drawn in dotted lines.*

noted GVG^2, is analogous to a GVG that is restricted to a two-equidistant face. 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; each of these graphs can be defined in terms of line of sight information.

Connectivity of the GVD. The underlying philosophy of the HGVG is to exploit the connectivity property of the GVD. Recall from Section 5 that the GVD is the union of all of 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 (a three-equidistant face) is formed by the intersection of *three* two-equidistant faces. By definition, a GVG edge exists 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 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 14. 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 some cases, additional links are required.

Second Order Generalized Voronoi Region. A GVG^2 on a two-equidistant face, $\mathcal{F}_{ij}$, denoted $\mathrm{GVG}^2\big|_{\mathcal{F}_{ij}}$, 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). See Figure 15. These regions are called *second order generalized Voronoi regions* and are defined as follows

$$\mathcal{F}_k\big|_{\mathcal{F}_{ij}} = \mathrm{cl}\{x \in \mathcal{F}_{ij} : \forall h \neq i,j,k,\; d_h(x) > d_k(x) > d_i(x) = d_j(x) > 0 \text{ and } \nabla d_i(x) \neq \nabla d_j(x)\}, \quad (14)$$

where d_i is the visible distance function, defined in Section 4.

It can 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). [9]

The rest of this section is now devoted to careful consideration of the structures on the boundary of a second order generalized Voronoi region. Furthermore, we show that these boundary structures are defined in terms of line of sight information and can be incrementally traced using range sensor data.

A rigorous proof enumerating all of the structures in the boundary of a second order generalized Voronoi region is contained in [9], but this result can be seen via inspection of Equation (14). The first inequality in Equation (14) is associated with the structures that exist on a common boundary of *two* second order generalized Voronoi regions on the same two-equidistant face. These structures are called the *second order two-equidistant face* (termed the GVG^2 equidistant edge in $\mathbb{R}^3$) and the *occluding two-face* (termed the occluding edge in $\mathbb{R}^3$).

GVG^2 Equidistant Edge. A GVG^2 equidistant edge, denoted $\mathcal{F}_{kl}\big|_{\mathcal{F}_{ij}}$, is the set of points on the boundary of adjacent second order generalized Voronoi regions where the distance to the second closest obstacle *continuously changes* as the robot crosses from one region to the other. In other words,

$$\mathcal{F}_{kl}\big|_{\mathcal{F}_{ij}} = \{x \in \mathcal{F}_{ij} \quad \text{such that} \\ \forall h,\, d_h(x) \geq d_k(x) = d_l(x) \geq d_i(x) = d_j(x) > 0\}. \tag{15}$$

The "equidistant" is included in this term to distinguish it from other GVG^2 edges such as occluding edges. In Figure 15, GVG^2 equidistant edges fully compose the GVG^2 on the two-equidistant face defined by the floor and ceiling.

Since the GVG^2 equidistant edge is defined in terms of the visible distance function, it can be incrementally constructed using only local range information. The incremental construction technique for GVG^2 equidistant edges is similar to that of GVG edges. That is, a GVG^2 equidistant edge is incrementally constructed by tracing the roots of

$$G_2(y, \lambda) = \begin{bmatrix} (d_1 - d_2)(y, \lambda) \\ (d_3 - d_4)(y, \lambda) \end{bmatrix}, \tag{16}$$

as the parameter λ is varied. In Equation 16, obstacles C_1 and C_2 are closest and obstacles C_3 and C_4 are second closest. Since the visible distance function makes up Equation (16), the incremental construction technique relies solely on line of sight information.

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. In $\mathbb{R}^3$, the second order three-equidistant face are *GVG^2 equidistant vertices*, or *second order meet points*.

Occluding Edge. The occluding edge is the set of points on the boundary of adjacent second order generalized Voronoi regions where the distance to the second closest obstacle does *not continuously change* as the robot crosses from one region to the other. See Example 2.

Boundary Edge, Floating Boundary Edge. The second inequality from Equation (14) is associated with a GVG edge (points where $d_i(x) = d_j(x) = d_k(x)$). Boundary edges (points where distance to the environment is zero) are associated with the next inequality and floating boundary edges (points where the gradients are collinear) are associated with the last inequality.

A key feature of all of these structures is that they can be incrementally constructed. A GVG edge can be incrementally traced using line of sight information (Section 6). Tracing a boundary edge is akin to wall following, and in a numerical context, it can be viewed as tracing the roots of

$$G_b(y, \lambda) = \begin{bmatrix} d_1(y, \lambda) \\ d_2(y, \lambda) \end{bmatrix}, \tag{17}$$

as the parameter λ is varied. Finally, it can be shown that a floating boundary edge is a straight line and need not require a complicated edge tracing procedure (but rather assumes the robot has a good dead reckoning sensor).

Connectivity. We have shown that the boundary of a second order generalized Voronoi region may contain: GVG edges, boundary edges, floating boundary edges, GVG^2 equidistant edges, and occluding edges. Now, we need to consider the connectivity of the boundary of a second order generalized Voronoi region.

In Figure 15 the GVG cycle is surrounded by a GVG^2 cycle which is connected to the outer GVG component. The existence of a cycle informs the robot that there is a need to make the link. The link is made by following the second closest obstacle distance gradient (or negated gradient) projected onto the tangent plane of two-equidistant face. With this link, the boundary of the second order generalized Voronoi region is connected.

In prior work [6], it is assumed that all GVG edges must have at least one meet point, and thus no GVG cycles can exist. This assumption is met in "cluttered" environments, which are the ones of most interest for sensor based planning. Nevertheless, even when

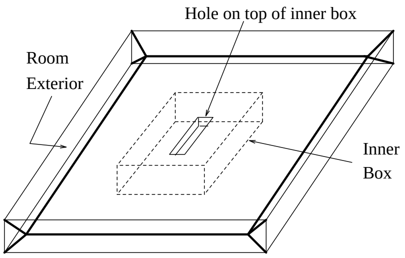

Figure 16: *Room with a box in the middle. The box, outlined with dotted lines, has an opening on top of it, delineated with solid lines.*

this assumption is met, there are two other scenarios in which the boundary of a second order generalized Voronoi region may not be connected: when there is an occluding cycle (a connected second order generalized Voronoi region boundary component that has only occluding edges), and when there is a boundary cycle (a connected second order generalized Voronoi region boundary component that has only boundary edges).

A cycle of occluding edges (as is the case in Example 2) is detected by looking for discontinuities in distance sensor readings as the robot traverses a boundary component of a second order generalized Voronoi region. Forming a link to (from) an occluding cycle is done via gradient ascent (descent) of D. A final linking procedure is required for cycles formed by boundary edges. A cycle of boundary edges is detected by looking for local maxima of distance sensor readings as the robot traverses a boundary component of a second order generalized Voronoi region. Forming a link to (from) an boundary cycle is also achieved via gradient ascent (descent) of D.

Since the boundaries of the second order generalized Voronoi regions are connected (or can be readily connected with a link), the $\mathrm{GVG}^2\big|_{\mathcal{F}_{ij}}$ connects disconnected GVG edges on a two-equidistant face. Since the GVD is connected, the GVG edges, combined with the GVG^2 edges and links, is connected. That is, the HGVG is connected.

Example 2 *Figure 16 is similar to Figure 14 except 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 17) contains two connected GVG components: one associated with the hole and ceiling, and one associated with the box, the floor, and the ceiling. Unfortunately, the two connected GVG components are not within line of sight of each other, and therefore an occluding edge is used to connect them.*

The GVG structure associated with the hole is connected to the occluding edge using GVG^2 equidistant edges. Using a linking procedure, the GVG cycle is linked to the occluding edge. Now, the GVG is connected through a link, an occluding edge, and an equidistant GVG^2 edge. See Figure 18.

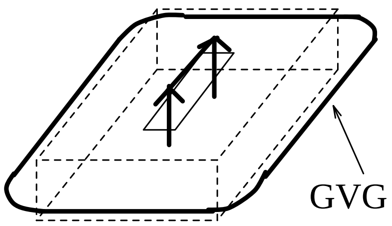

Figure 17: *Disconnected GVG components not within line of sight of each other.*

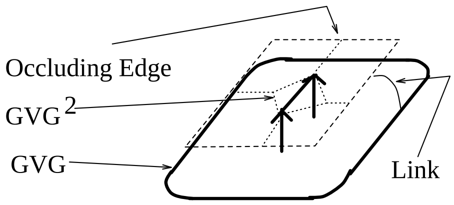

Figure 18: *Connected structure.*

Finally, it can be shown that all points in the environment are within line of sight of the HGVG, and thus the HGVG has the property of *departability.* This line of sight departabilty property of the HGVG indicates that the HGVG has applications to sensor placement (e.g., the art gallery problem).

9 Conclusion and Future Work

We have developed a rigorous basis for sensor based motion planning for a robot, modeled as a point.

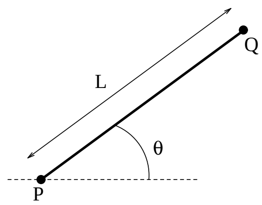

Figure 19: *The configuration of a rod is determined by the x and y coordinates of P and the orientation of the rod with respect to the horizontal.*

To this end, we defined the *hierarchical generalized Voronoi graph* (HGVG) to serve as a basis for robotic sensor based motion planning. The HGVG is a *roadmap*, which is a one-dimensional representation of an environment populated with obstacles, and has three key properties: (1) *accessibility,* (2) *connectivity,* and (3) *departability.* Simulations and experiments have validated this approach.

The ultimate goal of this work is to enable highly articulated robots equipped with sensors to explore unknown environments, via construction of a roadmap, and thus, many of the results of this work hold in spaces of dimension m. Nevertheless, the focus of this work is in dimension three where workspace distance measurements are available. Using these workspace distance measurements, the next step is to extend the definitions of the HGVG to the case of when the robot can be modeled as a line segment, sometimes called a rod (see Figure 19). The resulting roadmap is termed the *rod hierarchical generalized Voronoi graph* (rod-HGVG), the planar version of which has already been defined in [8].

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 20.) 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 6.2.

The next step is to extend the results of the rod roadmap to that of a convex set, which in turn will be extended to the development of a roadmap for a chain of convex sets which model a highly articulated robot.

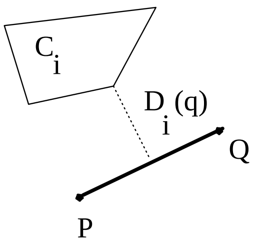

Figure 20: *The distance from the rod (thick solid line) to an obstacle is the distance (dotted line) between the nearest point on the rod to the obstacle and the nearest point on the obstacle to the rod.*

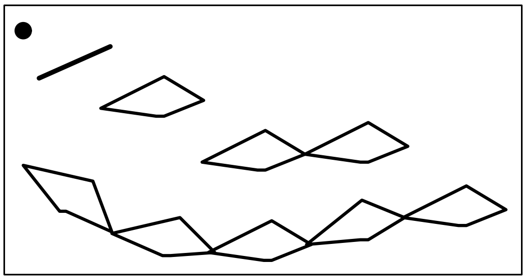

Figure 21: *Outline of future research.*

We also believe that our focus on workspace distance as the underlying geometric foundation of these roadmaps will better enable future work on the analysis of sensor error and noise on planning performance. A parallel direction of current research focuses on sensor limitations such as sensor noise, effective sensor range, and sensor quantization. (Sensor quantization considers the discretization of range data.) Future research will focus on the use of robot vision to generate GVG edges in environments where range data is not readily available.

References

[1] D. Avis and B.K. Bhattacharya. Algorithms for Computing d-dimensional Voronoi Diagrams and Their Duals. *Advances in Computing Research*, 1:159–180, 1983.

[2] R.A. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal on Robotics and Automation*, RA-2, March 1986.

[3] J.F. Canny and B. Donald. Simplified Voronoi Diagrams. *Discrete Comput. Geometry*, pages 219–236, 1988.

[4] J.F. Canny and M.C. Lin. An Opportunistic Global Path Planner. *Algorithmica*, 10:102–120, 1993.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Howie Choset. *Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. PhD thesis, California Institute of Technology, Pasadena, CA, 91125, 1996.

[10] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Society of Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[11] 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.

[12] H.B. Keller. *Lectures on Numerical Methods in Bifurcation Problems*. Tata Institute of Fundamental Research, Bombay, India, 1987.

[13] V. Lumelsky and A. Stepanov. Path Planning Strategies for Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape. *Algorithmica*, 2:403–430, 1987.

[14] C. Ó'Dúnlaing and C.K. Yap. A "Retraction" Method for Planning the Motion of a Disc. *Algorithmica*, 6:104–111, 1985.

[15] 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.

[16] 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.

[17] 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.

[18] P.F. Rowat. "Representing the Spatial Experience and Solving Spatial Problems in a Simulated Robot Environment". In *PhD. Thesis*, University of British Columbia, 1979.