Incremental Construction of the Generalized Voronoi Diagram, the Generalized Voronoi Graph, and the Hierarchical Generalized Voronoi Graph

Howie Choset

1 Overview

This work introduces a new *incremental* construction procedure for the *generalized Voronoi diagram* (GVD), the locus of points equidistant to two obstacles. This procedure is incremental in that it requires line of sight information (see Figure 4) to construct the GVD. Note that this notion of incremental is different from previous definition of incremental where the GVD is constructed by inserting one obstacle at a time into the environment. Finally, unlike other GVD construction techniques, this procedure does not place any special restrictions on the types of obstacles; that is, obstacles need not be polygons nor convex sets.

The incremental construction procedure was originally developed for sensor based planning of highly articulated robots. Sensor based planning incorporates sensor information into a robot's planning procedure, in contrast to classical planning which assumes full knowledge of the environment prior to planning. When a robot has no a priori information about the environment, the robot must employ an incremental motion planner because most environments do not contain one vantage point from which a robot can "see" the entire world, and thereby construct a plan or representation from such a single vantage point. Although the incremental construction procedure was designed for when the robot has no a priori knowledge of the environment, it is also useful when full or partial knowledge of the environment is available to the robot.

The GVD serves as a *roadmap* structure for a mobile robot operating in the plane. Roadmaps are geometric structures that capture the full topology of a robot's environment and have three key properties: accessibility, connectivity, and departability. Motion planning is achieved by planning a path onto the GVD (accessibility), planning a path in the GVD to the vicinity of the goal (connectivity) and then planning a path to the goal (departability). The GVD is useful in motion planning because the bulk of the motion planning occurs in a one-dimensional subset of the robot's two-dimensional environment. When a robot incrementally constructs the entire GVD for an environment, it has essentially explored that environment.

The incremental construction procedure is not limited to mobile robot applications, but rather it applies to a general class of highly articulated robots. Highly articulated robots are ones with many degrees of freedom; such robots include hyper-redundant manipulators (i.e., robot snakes) and cooperating robots. These robots tend to be modeled as a points in an m-dimensional configuration space and thus the GVD is not useful for these robots because the GVD is $(m-1)$-dimensional. Therefore, the *generalized Voronoi graph* (GVG) was introduced. The GVG is the one-dimensional set of points in m dimensions that are equidistant to m obstacles. The incremental construction procedure, reviewed below, generates the GVG (which is the GVD when $m = 2$).

Unfortunately, the GVG is not guaranteed to be connected, so additional structures, termed *higher order generalized Voronoi graphs*, are defined. The GVG, with these additional one-dimensional structures, is termed the *hierarchical generalized Voronoi graph* (HGVG). Again, when $m = 2$, the GVG = HGVG = GVD.

2 Voronoi Diagrams

The Voronoi diagram is a geometric structure that has received much attention in recent years in many fields, including biology, computer science, city planning, crystallography, and robotic motion planning, which was the motivation of this work. The Voronoi diagram sections off a space into regions closest to a particular point, called a site. Therefore, the Voronoi diagram is the collection of hyper-planar patches that are equidistant to two sites such that the two sites are than any other site to the hyper-planar patch. See [2] for an extensive survey of Voronoi diagrams.

An important generalization of the Voronoi diagram considers non-point sites; such a structure is termed a *generalized Voronoi diagram* (GVD), which was first used by [18] in a digital environment. Active research in applying the GVD to motion planning began with Ó'Dúnlaing and Yap's work in [16], which considered motion planning for a disk in the plane.

The GVD is sometimes called a *retract.* Let the freespace, $\mathcal{FS}$, be the set of points not occupied by obstacles, and $Im\colon \mathcal{FS} \rightarrow GVD$ be a continuous function termed a *retraction* which maps every point in the freespace onto the GVD. The GVD is a retract when Im can be continuously deformed to the identity mapping, that is, there exists a homotopy function $f\colon \mathcal{FS} \times [0,1] \rightarrow GVD$ such that $f(x,0) = x$, $f(x,1) = Im(x)$, and $f(a,t) = a$ for all $a \in \mathcal{FS}$ and

---

Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

for all $t \in [0, 1]$. The GVD, which is a retract, captures all of the geometries of the free space.

One-dimensional retracts are members of a general class of structures, termed *roadmaps* (Canny, [3]), which are a collection of one-dimensional curves that capture the important topological and geometric properties of a robot's environment. Roadmaps are analogous to highway systems and have the following properties: *accessibility*, *connectivity*, and *departability*. Using a roadmap, 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). An example of a complete roadmap scheme is Canny and Lin's Opportunistic Path Planner (OPP) [4].

Previous work (Choset and Burdick, [10]) introduces a roadmap termed, *the hierarchical generalized Voronoi graph* (HGVG), an extension of the GVD into higher dimensions. The backbone of the HGVG is the *generalized Voronoi graph* which is the locus of points equidistant to m obstacles in an m dimensional space. The GVG is one dimensional, whereas the GVD is co-dimension one. However, unlike the GVD, the GVG is not necessarily connected in dimensions greater than two, and thus, in general, is not a roadmap. Additional structures, termed *higher order generalized Voronoi graphs*, which when used in conjunction with the GVG, form a connected network. The resulting connected structure is the HGVG.

A key feature of the HGVG is that it is defined in terms of line of sight distance measurements. These definitions apply to the GVG and GVD and appear to be a new way of defining these structures.

An advantage that the HGVG has over other roadmaps is that it possesses an incremental construction procedure. In this work, "incremental" takes on a different meaning from previous Voronoi diagram work. For example, Fortune [13] developed a "sweepline" approach where a deformation of the Voronoi diagram is computed. The deformed Voronoi digram is computed in an incremental manner in that a sweepline passes through a deformed space to construct the deformed diagram. Once the deformed diagram is computed, the Voronoi diagram is determined. Another type of incremental approach computes the Voronoi diagram by adding one site at a time.

In this work, "incremental" means constructing a portion of the Voronoi diagram (or its generalizations into higher dimensions) using only line of sight information. One such incremental approach which creates Voronoi Diagram-like structures can be found in [17], but it is restricted case of polygonal obstacles deployed in planar environments.

Recently, Choset and Burdick [9] introduced a new method for incrementally constructing the GVD, GVG, and HGVG. This procedure relies solely on line of sight information to generate the appropriate graph edges. An advantage of this approach is that it only requires distance information to the obstacles that are within line of sight of the robot. This procedure does not place any special restrictions on the types of obstacles (polyhedral, convex, etc.) – in fact, the obstacles can be specified as fuzzy sets. (It should be noted that Yap [20] prescribed a Voronoi diagram construction algorithm for a planar world with curved surfaces.)

3 Distance Functions

Since the original motivation of this work is complete sensor based planning for robots, we consider the distance function because it encodes the distance between a robot and a nearby obstacle. This function is key to our subsequent definitions and results and allows us to generate Voronoi diagrams of any type of obstacle: point obstacle, polygonal obstacles, convex obstacles, concave obstacles, fuzzy obstacles, etc.

3.1 X-Distance Function

Definition 3.1 (Single Object Distance Function) *The distance between a point, x and a convex set C_i is distance between the point x and closest point on the obstacle C_i, i.e.,*

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

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

In Clarke [12] it is shown that the gradient of $d_i(x)$ is

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

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 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). 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 points in the direction of the negated distance gradient.

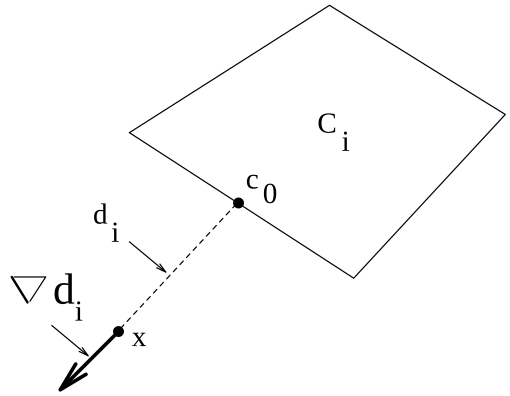

Fig. 1. Distance between x and C_i is the distance to the closest point on C_i. The gradient is a unit vector pointing away from the nearest point.

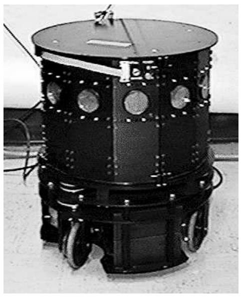

Fig. 2. Mobile robot with sonar ring.

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

Definition 3.2 (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*

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

It is shown in [7] that the multi-object distance function is nonsmooth, and hence its gradient cannot be trivially defined. However, using nonsmooth analysis (which is reviewed in [7]), it can be shown that the *generalized gradient* of $D(x)$ is

$$\partial D(x) = \mathrm{Co}\{\nabla d_i(x) \,:\, i \in I(x)\}, \tag{4}$$

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 3. Notationally, if ∂ appears in front of a set, as opposed to a function, then it means the boundary of the set.

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.

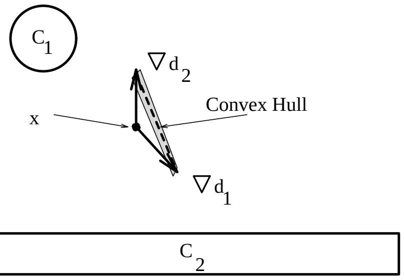

Fig. 3. The generalized gradient of the multi-object distance function at a point is the convex hull of the gradients of the single object distance functions which correspond to the closest equidistant obstacles at that point. The solid arrows are single object distance gradients and the shaded region corresponds to the heads of all of the vectors which are in the convex hull of the two single object gradients.

3.2 Visible Distance Function

The approach summarized in this work only relies on line of sight information and thus the X-ray distance function must be adapted for realistic sensor based implementation. 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.,

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

Let c be the nearest point in C_i to x, as defined by the X-distance function (i.e., $c =$ 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.

Definition 3.3 (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.,*

$$d_i(x) = \begin{cases} \min_{c \in C_i} \|x - c\|, & \text{if } c \in \mathrm{int}(\tilde{C}_i(x)), \\ \infty, & \text{if } c \notin \mathrm{int}(\tilde{C}_i(x)). \end{cases} \tag{5}$$

See Figure 4 for an example of the visible distance function.

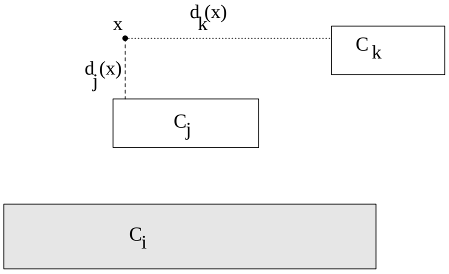

Fig. 4. Obstacle C_i is occluded from x because the shortest straight line segment between x and C_i penetrates obstacle C_j.

We define all structures in this work in terms of the visible distance function, which relies solely on line of sight information.

4 Generalized Voronoi Graph

Using the above described distance functions, the HGVG can be defined. First, the GVD and GVG are defined in terms of the distance function. The basic building block of the GVD and GVG is locus of points equidistant to two obstacles, C_i and C_j (i.e., $d_i(x) = d_j(x)$), such that the distance is closer to these two obstacles than to all other obstacles ($\forall h\ d_h(x) \geq d_i(x)$). This structure is termed the *two-equidistant face*,

$$\mathcal{F}_{ij} = \text{cl}\{x \in \mathcal{W}\backslash(C_i \bigcup C_j) : \\ \forall h\ d_h(x) \geq d_i(x) = d_j(x) \geq 0 \text{ and } \nabla d_i(x) \neq \nabla d_j(x)\}. \tag{6}$$

See Figure 5. Note that the distance gradients to the two closest obstacles are distinct ($\nabla d_i(x) \neq \nabla d_j(x)$) which allows non-convex sets to be defined as the finite union of convex sets.

The *two-Voronoi set*, $\boldsymbol{\mathcal{F}}^2$, is the union of all two-equidistant faces, i.e. $\boldsymbol{\mathcal{F}}^2 = \bigcup_{i=1}^{n-1} \bigcup_{j=i+1}^{n} \mathcal{F}_{ij}$. Since $\boldsymbol{\mathcal{F}}^2$ is the set of points equidistant to the two or more closest points on the boundary of $\mathcal{W}$, it is the generalized Voronoi diagram (GVD) of $\mathcal{W}$. See Figure 5. This definition of the GVD appears to be new and does not place any restriction on the type of obstacles which populate the environment.

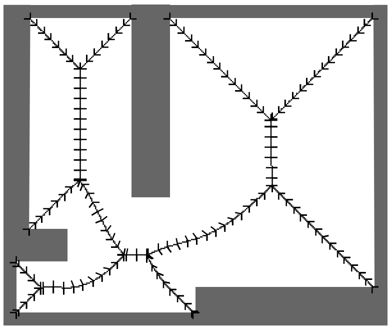

Fig. 5. The ticked solid lines is the set of points equidistant to two obstacles, such that each edge fragment is closest to the equidistant obstacles.

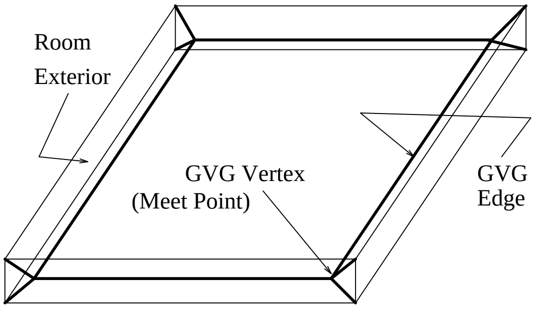

Fig. 6. The generalized Voronoi graph in a rectangular enclosure. The solid lines represent the GVG edges.

The pre-image theorem asserts that $\boldsymbol{\mathcal{F}}^2$ (i.e., the GVD) is $(m-1)$-dimensional [8]. The GVD does reduce the motion planning problem by a dimension, but a one-dimensional roadmap is required. Observe that the two-equidistant faces, $\mathcal{F}_{ij}$, $\mathcal{F}_{ik}$, and $\mathcal{F}_{jk}$ intersect to form an $(m-2)$-dimensional manifold that is equidistant to three obstacles. Such a structure is termed a *three-equidistant face* and is denoted $\mathcal{F}_{ijk}$. This intersection procedure is repeated until a one-dimensional structure is formed; such a structure is an m-equidistant face, $\mathcal{F}_{i_1 \ldots i_m}$ and is a one-dimensional set of points equidistant to m objects in m dimensions. (Also note, an $m+1$-equidistant face is formed in a similar way and is always a point.)

The *generalized Voronoi graph* (GVG) is the collection of m-equidistant faces and $m+1$-equidistant faces, i.e., GVG = $(\boldsymbol{\mathcal{F}}^m, \boldsymbol{\mathcal{F}}^{m+1})$. The m-equidistant faces are termed *GVG edges* and $m+1$-equidistant faces are termed *meet points.* Note that the GVD is $m-1$-dimensional whereas the GVG one-dimensional. Also, the GVD is the locus of points equidistant to *two* obstacles whereas the GVG is the locus of points equidistant to m obstacles. (See Figure 6 for an example of a generalized Voronoi graph for a rectangular enclosure in $\mathbb{R}^3$ where 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.)

5 Transversality: No four points are cocircular

In order to use the pre-image theorem to determine the generic dimension of the GVG edges (and higher dimensional equidistant faces), we first introduce an important transversality assumption and discuss its implications.

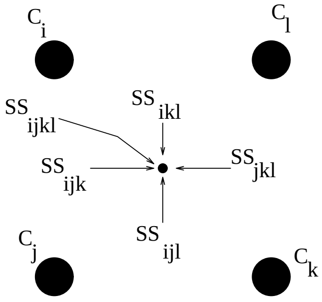

Fig. 7. Non-generic arrangement.

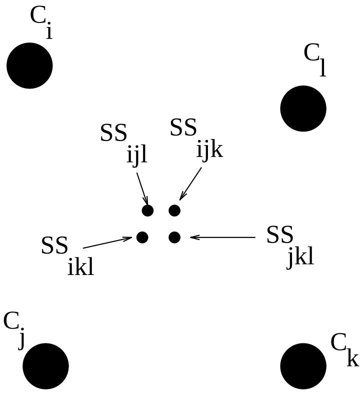

Fig. 8. Small perturbation in obstacle locations.

Assumption 5.1 (*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 5.1 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 Figure 7, $\mathcal{SS}_{ijk} = \mathcal{SS}_{jkl} = \mathcal{SS}_{ikl} = \mathcal{SS}_{ijl}$ 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 8). 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 5.1, $\mathcal{SS}_{i_1 \ldots i_k j_1} \neq \mathcal{SS}_{i_1 \ldots i_k j_2}$ 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 7, $\mathcal{SS}_{ijk} = \mathcal{SS}_{jkl} = \mathcal{SS}_{ikl} = \mathcal{SS}_{ijl}$, but after a slight perturbation of C_i, all the equidistant faces are no longer coincidental and $\mathcal{SS}_{ijkl} = \emptyset$.

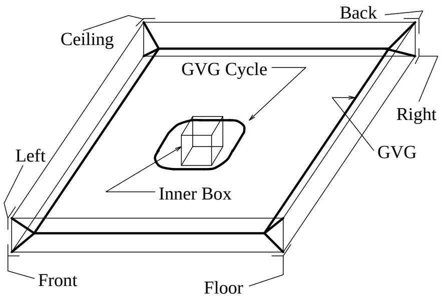

Fig. 9. Disconnected GVG.

6 Hierarchical Generalized Voronoi Graph

In the planar case, the GVG and GVD are the same, so the GVG is connected [16]. However, in higher dimensions, the GVG is not necessarily connected, as can be seen in Figure 9 which contains an example of a disconnected GVG with two connected components: (1) an outer GVG network similar to the one in Figure 6 and (2) an inner GVG network that forms a halo-like structure around the inner box. Note that in $\mathbb{R}^3$, GVG edges are in the boundary of two-equidistant faces. The GVG in Figure 9 is disconnected because the boundary of the two equidistant face, defined by the floor and ceiling in Figure 9, has a disconnected boundary.

The generalized Voronoi regions and equidistant faces may be viewed as a cellular decomposition of $\mathcal{W}$ into k-dimensional sets, 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 which is connected [19]. The two-equidistant face, defined by the floor and ceiling in Figure 9, is not homeomorphic to a two-dimensional disk because it has a hole in the middle of it, and thus the one-dimensional cells (i.e., the GVG) are not connected.

In higher dimensions, additional structures, termed *higher order generalized Voronoi graphs*, must be constructed to connect GVG components. Essentially, higher order generalized Voronoi graphs are like GVG's that are recursively defined on lower dimensional 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. 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 are defined in terms of the distance function.

For the moment, consider only the three-dimensional case in which the only higher order generalized Voronoi

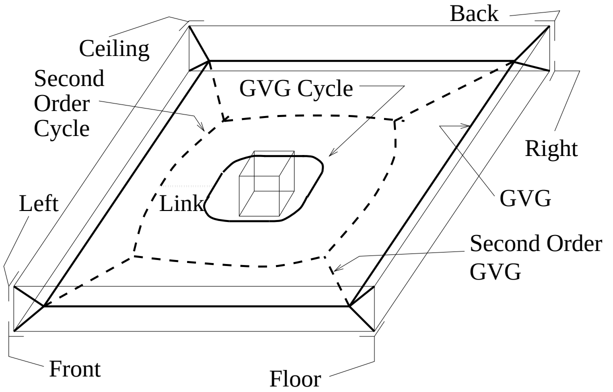

Fig. 10. The GVG^2 is drawn in dotted lines.

graph is the GVG^2 which is defined on a two-equidistant face. One of the structures in the GVG^2 is termed a *GVG^2 equidistant edge*, and is denoted $\mathcal{F}_{kl}\big|_{\mathcal{F}_{ij}}$. It is the set of points whose closest pair of equidistant obstacles are C_i and C_j and whose second pair of equidistant obstacles are C_k and C_l, i.e.,

$$\begin{aligned}\mathcal{F}_{kl}\big|_{\mathcal{F}_{ij}} &= \{x \in \mathcal{F}_{ij} \quad \text{such that} \\ \forall h,\ d_h(x) &\geq d_k(x) = d_l(x) \geq d_i(x) = d_j(x) > 0\}. \end{aligned} \tag{7}$$

See Figure 10 for examples of GVG^2 equidistant edges. Note that in Figure 10 there is a duality between the GVG cycle (the halo-like structure surrounding the box) and the GVG^2 cycle. In most situations, this relationship allows the robot to make well defined links that connect the GVG and GVG^2 fragments. The result is a connected roadmap [8], [10], which is the HGVG.

7 Connectivity: The Generalized Voronoi Complex

The proof of connectivity of the GVD relies on continuity of the retraction function, Im. Continuity is important because the image of a connected set under a continuous function is a connected set [1]. The image of a connected component of free space under the continuous function Im is the GVD, and thus the GVD is connected.

Unfortunately, there is no continuous function whose image is the HGVG, and thus the proof of connectivity of the HGVG cannot use the same continuity argument. Instead, a new structure, termed the *generalized Voronoi complex*, is introduced and proven to be a connected structure. Proof of connectivity of the HGVG is reduced to showing that the HGVG is a generalized Voronoi complex, which is done in [11]. This proof methodology does not require as much structure as previous approaches.

7.1 Notation

The following notation is used in defining the generalized Voronoi complex.

Definition 7.1 (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 7.2 (Adjacency) *Two sets A_i and A_j are* adjacent *if*

$$\text{cl}(A_i) \bigcap \text{cl}(A_j) \neq \emptyset.$$

Definition 7.3 (Sub-adjacency) Two sets, A_i and A_j, are *sub-adjacent* if

$$\dot{\partial} A_i \bigcap \dot{\partial} A_j \neq \emptyset.$$

7.2 Definitions

The generalized Voronoi complex is an exact cellular decomposition of generalized Voronoi cells.

Definition 7.4 (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, the set of points closest to one obstacle, are also examples of generalized Voronoi cells.

The generalized Voronoi cells comprise the *generalized Voronoi complex.*

Definition 7.5 (Generalized Voronoi Complex in X) *The* generalized Voronoi complex, *$\boldsymbol{\mathcal{V}}^2$, of a connected set X is the union of the boundaries generalized Voronoi cells, $\bigcup_i \dot{\partial}\mathcal{V}_i$, such that the generalized Voronoi cells form an exact cellular decomposition of X and adjacent generalized Voronoi cells are sub-adjacent:*

1. *$\bigcup_i \mathcal{V}_i = X$,*
2. *$\text{int}(\mathcal{V}_i) \bigcap \text{int}(\mathcal{V}_j) = \emptyset \quad \forall i, j$,*
3. *$\text{cl}(\mathcal{V}_i) \bigcap \text{cl}(\mathcal{V}_j) \neq \emptyset \iff \dot{\partial}\mathcal{V}_i \bigcap \dot{\partial}\mathcal{V}_j \neq \emptyset$.*

7.3 Examples

It is shown below that the generalized Voronoi complex is a connected structure. For example, the GVD is a generalized Voronoi complex because generalized Voronoi regions are generalized Voronoi cells, and the set of generalized Voronoi regions in a connected component of free space satisfy the above listed criteria. Therefore, the union of the boundaries of the generalized Voronoi regions forms a generalized Voronoi complex. Hence, the GVD (the union of the boundaries of the generalized Voronoi regions) is a generalized Voronoi complex.

The trapezoidal decomposition [15] forms another example of a generalized Voronoi complex. By construction, each of the trapezoids in the trapezoidal decomposition has a connected sub-boundary and adjacent cells share a common sub-boundary component. Finally, by construction, the union of the cells is the space, itself. Therefore, the union of the boundaries of the trapezoids forms a connected structure.

Note that the above two examples hint at the relationship between the cellular decomposition and roadmap approaches in robot motion planning. A special case of this relationship is discussed in [19] where the cells are all homeomorphic to k-dimensional disks.

7.4 Proof of Connectivity

PROPOSITION 7.6 *The generalized Voronoi complex is connected.*

Proof: Let s and g be the end points (start and goal) of a path fully contained in a generalized Voronoi cell, $\mathcal{V}_i$. Since there are no obstacles in the interior of $\mathcal{V}_i$, there exists a path, c_1, from s to a point, s^*, in $\dot{\partial}\mathcal{V}_i$ and there exits a path from g to a point, g^*, in $\dot{\partial}\mathcal{V}_i$. By definition of a generalized Voronoi cell, there is a path between s^* and g^* in $\dot{\partial}\mathcal{V}_i$. Therefore, for the case where there is one cell in a generalized Voronoi complex, the generalized Voronoi complex is connected because s and g are arbitrarily chosen.

Assume that a path, c_n, passes through $\mathcal{V}_{i_1}, \ldots, \mathcal{V}_{i_n}$, which are n generalized Voronoi cells whose sub-boundaries form a connected set. Let s and g be the end points of a new path, c_{n+1}, where $s \in \mathcal{V}_{i_1}$ and $g \in \mathcal{V}_{i_{n+1}}$. The path c_{n+1} prescribes a sequence of adjacent generalized Voronoi cells through which the c_{n+1} passes. Therefore, $\mathcal{V}_{i_n}$ and $\mathcal{V}_{i_{n+1}}$ are adjacent and by definition of a generalized Voronoi complex, the $\dot{\partial}\mathcal{V}_{i_n} \bigcup \dot{\partial}\mathcal{V}_{i_{n+1}}$ is a connected set. Therefore, union of the sub-boundaries of $\mathcal{V}_{i_1}, \ldots, \mathcal{V}_{i_n}, \mathcal{V}_{i_{n+1}}$ forms a connected set.

Since s and g are arbitrary, by induction the generalized Voronoi complex is connected.

■

It can be shown the HGVG is a generalized Voronoi complex [11], and thus is a connected structure.

8 Incremental Construction of the HGVG

Unlike other motion planners, line of sight information suffices to provide the incremental construction procedure that directs the robot to access the HGVG, to trace the HGVG edges, and then to depart from the HGVG. After the robot incrementally constructs the HGVG, it has in essence explored an environment. For the purposes of explanation, we describe first the traceability procedure, followed by accessibility and departability.

8.1 Traceability

In an incremental context, the property of connectivity is interpreted as *traceability*, which implies that, using only local data, the robot can: (1) "trace" the HGVG edges; (2) determine when to terminate the edge tracing process, and (3) determine when to start new edge tracing procedures. For the sake of discussion, this section is limited to tracing GVG edges, but the results of this section are easily generalized to all HGVG components.

The GVG incremental approach to edge construction borrows ideas from numerical continuation methods [14], which trace the roots of the expression $G(y, \lambda) = 0$ as the parameter λ is varied. For the case of the GVG, the tracing function $G\colon \mathbb{R}^{m-1} \times \mathbb{R} \to \mathbb{R}^{m-1}$ is

$$G(y, \lambda) = \begin{bmatrix} (d_1 - d_2)(y, \lambda) \\ (d_1 - d_3)(y, \lambda) \\ \vdots \\ (d_1 - d_m)(y, \lambda) \end{bmatrix} \tag{8}$$

The function $G(y, \lambda)$ assumes a zero value only on a GVG edge. Hence, if the Jacobian of G is invertible, then the implicit function theorem implies that the roots of $G(y, \lambda)$ locally define a GVG edge as λ is varied. A GVG edge is constructed by numerically tracing the roots of G.

The explicit edge construction procedure has two steps: a predictor step and a corrector step. The predictor step moves the robot for a small distance along the tangent of the GVG. This tangent is the vector orthogonal to the m closest points in the m closest obstacles [9]; note that the m closest points in the m closest obstacles are within line of sight of the robot. Typically, the prediction step takes the robot off of a GVG edge, so a correction procedure is required to bring the robot back to the GVG. If step size along the tangent is "small," then the graph will intersect a "correcting plane" (Figure 11), which is a plane orthogonal to the tangent. The correction step finds the location where the GVG intersects the correcting plane (Figure 11) and is achieved via 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

$$y^{k+1} = y^k - \left(\nabla_y G\right)^{-1} G(y^k, \lambda^k) \tag{9}$$

where $\nabla_y G$ is the Jacobian of G evaluated at (y^k, λ^k). It is shown in [9] that $\nabla_y G$ is invertible and thus Equation (9) is well posed. Practically speaking, this result states that the numerical procedure defined by Equation (9) will be robust for reasonable errors in robot position, sensor errors, and numerical roundoff.

The explicit terminating conditions for edge tracing include meet points and boundary points, locations

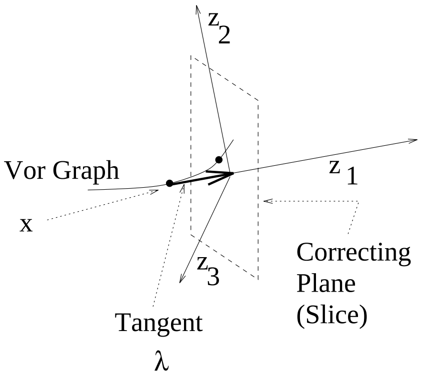

Fig. 11. Sketch of Continuation Method

Fig. 12. Experiment.

where a GVG edge intersections m obstacles [9], [10]. Incremental construction of the GVG is akin to a graph search where the meet points and boundary points are the GVG "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. If the robot reaches another unvisited meet point, the above procedure is recursively repeated. When the robot hits a boundary point, its marks the edge as explored. Then, the robot 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.

The results of this section can be generalized to the other HGVG structures by simply adjusting the G function (Equation 8). An experimental result where a mobile robot constructs a GVG is shown in Figure 12

8.2 Incremental Accessibility

In [16], accessibility is achieved in two steps (1) draw a straight line between the location of the robot and the closest point on the closest obstacle, and (2) move along this line away from the closest point. Essentially, this procedure defines Im, the retraction mapping that defines the GVD.

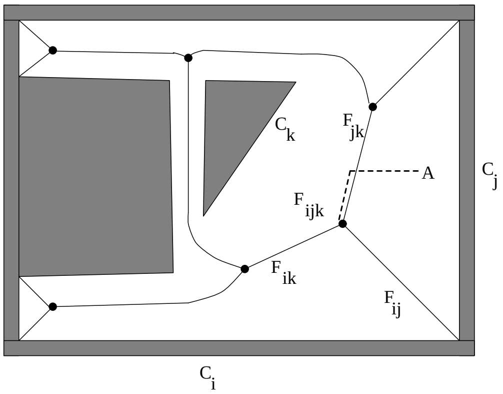

Fig. 13. Gradient ascent accessibility in $\mathbb{R}^3$.

Unfortunately, there is no retraction which maps the freespace onto the HGVG. Instead, the robot can access the GVG (and thus the HGVG) by following a path that is constructed using a sequence of gradient ascent operations of the multi-object distance function $D(x)$. Assuming the robot starts at a point closest to a particular object, the robot increases its distance away from its closest object until it is equidistant to two objects. Next, while maintaining this "double equidistance," the robot moves away from its two closest obstacles until it is the same distance from a third object (i.e., until it is triple equidistant). The robot repeats this procedure until it is m-wise equidistant, i.e., until it accesses the GVG.

Note that this procedure reduces to the draw-the-straight-line technique when $m = 2$.

Example 8.1 Figure 13 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. ♦

Whereas the previous section describes how to trace a path along a GVG edge, the remainder of this section describes how to trace a path on a k-equidistant face during the accessibility procedure. Recall that in Section 8.1, discussion focuses only on tracing the GVG edges because the remaining HGVG structures can be generated by appropriately adapting the G trace function. In this section, we derive the appropriate G function to trace a path on a k-equidistant face and use it in the numerical continuation methods [14] in a fashion similar to the approach described in Section 8.1.

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 $G_A : \mathbb{R}^{k-1} \times \mathbb{R}^{m-k+1} \to \mathbb{R}^{k-1}$ as follows:

$$G_A(y, \lambda) = \begin{bmatrix} (d_1 - d_2)(y, \lambda) \\ (d_1 - d_3)(y, \lambda) \\ \vdots \\ (d_1 - d_k)(y, \lambda) \end{bmatrix}. \tag{10}$$

Assume 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 8.1. If y^h and λ^h are the hth estimates of y and λ, the $h+1$st iteration is defined as

$$y^{h+1} = y^h - \left(\nabla_y G_A\right)^{-1} G_A(y^h, \lambda^h) \tag{11}$$

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.

The predictor step is a small step in the tangent space of the k-equidistant face which maximally increases $D(x)$. This step is determined in two stages: first the tangent space of the k-equidistant face at x is computed, and then the generalized gradient of D is projected onto it. The tangent space is the $(m - k + 1)$-dimensional plane that is orthogonal to the plane which contains the k closest points [11].

Now, it needs to be determined how the generalized gradient of D projects onto the k-equidistant face. The following proposition asserts that the generalized gradient of D, constrained to a k-equidistant face is smooth (even though D in the ambient space is not). This means, the generalized gradient of D projects onto a unique vector in the tangent space of each point on a k-equidistant face and this vector is given by the following proposition, whose proof appears in [11].

PROPOSITION 8.2 *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:

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

The correction procedure for the k-equidistant face is identical to the correction procedure for the GVG edge, except the correcting plane is $k - 1$-dimensional, not $m - 1$ dimensional. In [11], the Jacobian of G_A is shown to be invertible.

8.3 Departability

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.*

9 Conclusion and Future Work

This paper reviews the Voronoi diagram aspects of recent work in sensor based planning that appears in [8], [9], [10], [11]. This recent work includes the definition of a structure, termed the *hierarchical generalized Voronoi graph* (HGVG), and the prescription of the structure's incremental construction procedure. The HGVG is a generalization of the GVD into higher dimensions and is useful in motion planning because the HGVG is a *roadmap* structure. Motion planning is achieved by planning a path onto the roadmap (accessibility), planning a path in the roadmap to the vicinity of the goal (connectivity) and then planning a path to the goal (departability). Motion planning with the HGVG in an m-dimensional Euclidean space is useful because the bulk of the motion planning occurs on the one-dimensional HGVG. This structure was originally developed for motion planning of hyper-redundant manipulators [5], [6], but the HGVG can be used for motion planning of a broad class of robots and the HGVG can be applied to non-robotic problems such as injection molding.

A key feature of the HGVG is its incremental construction procedure that requires only line of sight information. Since line of sight information is the upper-bound to what sensors readily provide, the incremental construction procedure is well suited to sensor based planning of autonomous robots whose configuration spaces are multi-dimensional.

A novel aspect of the incremental construction procedure is that it places no special restrictions on the types of obstacles. That is, this procedure does not care if the obstacles are point sites, convex sets, concave sets, curved sets, fuzzy sets, etc. In the case when $m = 2$, the HGVG is the generalized Voronoi diagram, and thus this procedure is also a new method for constructing generalized Voronoi diagrams in the plane.

Future work includes adapting the incremental construction procedure to construct the two-dimensional Voronoi diagram in $\mathbb{R}^3$. The current techniques can trace the boundary of the two-dimensional Voronoi sheets and new methods are required to trace the knot points (to be used in a a mesh or triangulation) on the sheets themselves. Such an approach will be an adaptation of the accessibility path tracing technique described in Section 8.2 which can be used to trace a path on a two-dimensional two-equidistant equidistant face.

References

[1] R. Abraham, J.E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications.* Springer-Verlag, New York, 2 edition, 1988.

[2] F. Aurenhammer. Power Diagrams: Properties, Algorithms, and Applications. *SIAM J. Comput.*, 16:78–96, 1987.

[3] J.F. Canny. *The Complexity of Robot Motion Planning.* MIT Press, Cambridge, MA, 1988.

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

[5] G.S. Chirikjian and J.W. Burdick. Kinematically Optimal Hyper-Redundant Manipulator Configurations. *IEEE Trans. on Automation and Robotics*, December 1995.

[6] G.S. Chirikjian and J.W. Burdick. The Kinematics of Hyper-Redundant Robotic Locomotion. *IEEE Trans. on Automation and Robotics*, December 1995.

[7] 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.

[8] 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.

[9] 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.

[10] H. Choset and J.W. Burdick. Sensor Based Planning: The Hierarhical Generalized Voronoi Graph. In *Proc. Workshop on Algorithmic Foundations of Robotics*, Toulouse, France, 1996.

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

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

[13] S. Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2:153–174, 1987.

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

[15] J.C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991.

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

[17] 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.

[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.

[19] 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.

[20] C.K. Yap. An $o(n\log n)$ Algorithm for the Voronoi Diagram of a Set of Simple Curve Segments. *Discrete Computation Geometry*, 2:365–393, 1987.