To appear in 1995 IEEE International Conference on Robotics and Automation

Sensor Based Planning, Part II: Incremental Construction of the Generalized Voronoi Graph

Howie Choset Joel Burdick
Division of Engineering and Applied Science
California Institute of Technology, Pasadena, CA 91125

Abstract: *This paper prescribes an incremental procedure to construct the* **Generalized Voronoi Graph** *(GVG) and the* **Hierarchical Generalized Voronoi Graph** *(HGVG) detailed in the companion paper [4]. The procedure requires only local distance sensor measurements, and therefore the method can be used as a basis for sensor based planning algorithms.*

1 Introduction

In [4] we introduced a set of 1-dimensional curves which we termed the Generalized Voronoi Graph (GVG), and its extension, the Hierarchical Generalized Voronoi Graph (HGVG). These curves are *retract-like,* in that they have the useful properties of accessibility and departability. Furthermore, under conditions stated in the companion paper, the HGVG is connected, and thus can form the basis for a complete motion planning scheme. We assume that the reader is familiar with the definitions and results in the companion paper [4].

In this paper, we develop methods to *incrementally* construct the GVG and HGVG. It is worth noting that the incremental construction procedure can be used to construct the retract-like network when full geometry of the world is available. More importantly, we show how to implement the incremental construction technique using only local distance sensor data. Hence, this method can form the basis for sensor based planning schemes that connect two points in a robot's free space, or that build concise "maps" which encode topological information about a robot's free space.

We briefly review other relevant work in sensor based or incremental planning schemes. Many sensor based schemes have been developed, mostly for the planar case (see [8] for a review of planar sensor based planning). However, very little analysis of the numerical properties of these schemes has been done. This paper presents a more thorough analysis of the numerical properties of our algorithm than is typical in the sensor based planning literature. Canny and Lin's OPP [1] constructs part of its roadmap (the freeways) using local information, and is therefore partially incremental. However, the construction of "bridge curves," which guarantee the roadmap's connectivity, requires the identification of "interesting critical points." Complete prior knowledge of the world's geometry is needed to identify the critical points. This is a major limitation of their algorithm for sensor based implementation. Rimon and Canny [7] have recently suggested a way to "sensorize" the OPP algorithm. They introduce the notion of a "critical point sensor," though the implementation of such a sensor is not well detailed. Furthermore, they do not provide a rigorous way to construct the freeway segments from sensor data. In contrast, this paper formulates a methodical construction of the roadmap segments from sensor data. One incremental approach which creates Voronoi Diagram-like structures can be found in [6], but it is restricted to the planar case.

The GVG's and HGVG's properties of accessibility, departability and connectivity translate to *incremental accessibility, incremental departability,* and *traceability,* respectively, in the incremental construction of the GVG. The remainder of this paper describes how to move onto (incremental accessibility), trace along (traceability), and depart from (incremental departability) the GVG *using only local information.* The algorithm is verified by experiments that are reviewed in Section 6. In the sequel, we focus on the incremental development of the GVG. The extensions of these incremental techniques to the HGVG is highly analogous to that of the GVG.

2 Incremental Accessibility

We define *Incremental Accessibility* to be the ability to access some point of the GVG via a collision free path from any point in the free space, using only *local information.* Incremental Accessibility is obtained by a sequence of gradient ascent operations of the Multi-object Distance function, defined below.

Definition 2.1 (Multi-Object Distance Function) The distance between a point x and the set of all obstacles $\mathcal{C}_i$, $i = 1, \cdots, n$ in the environment is defined as:

$$D(x) = \min_i d_i(x). \tag{1}$$

That is, the distance between point x and the environment is considered to be the distance to the nearest obstacle. Using nonsmooth analysis (which is reviewed in [2]), it can be shown that the *generalized gradient* of $D(x)$ is

$$\partial D(x) = co\{\nabla d_i(x) \,:\, i \in I(x)\}, \tag{2}$$

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

By performing a sequence of gradient ascent operations on the multi-object distance function, D, along Equidistant Faces, the robot can travel via a collision-free path from any point in the free space to the GVG. Assuming that the robot's initial configuration does not lie on an Equidistant Face, the robot first performs gradient ascent on D (whose gradient will be ∇d_{i_1} for some i_1) until it reaches $\mathcal{F}_{i_1 i_2}$. From here, the robot performs gradient ascent on D, but constrained to $\mathcal{F}_{i_1 i_2}$, until it reaches a triple equidistant face, $\mathcal{F}_{i_1 i_2 i_3}$. This process is repeated until the robot is equidistant to m objects, and therefore lies on some $\mathcal{F}_{i_1 \ldots i_m}$– an edge of the GVG.

Example: Figure 1 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. In one example, starting from (A), the robot follows gradient ascent of d_j until it reaches $\mathcal{F}_{jp}$. From there, it does gradient ascent of $D = d_p = d_j$ constrained to $\mathcal{F}_{jp}$ until it reaches $\mathcal{F}_{ijp}$, an edge of the GVG. There are two important things to note about this Incremental Accessibility procedure. First,

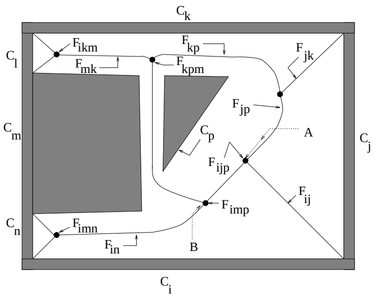

Fig. 1. Gradient ascent accessibility in 3-D

the procedure is based on a gradient ascent operation that is constrained to an equidistant face. The constrained gradient was analyzed in [2], where it was shown that the proper gradient can be obtained by a projection of the multi-object function generalized gradient (which is obtained directly from sensor data) onto the equidistant face tangent space. Second, it was shown in [2] that D is nonsmooth at local maxima, and so special care must be taken in terminating the gradient ascent operation. See [2] for details.

3 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 meet at a Generalized Voronoi Vertex; (3) change directions at a vertex, and thereby begin tracing new edges; and (4) determine when to terminate the tracing procedure. In this section, we present and analyze a method for tracing a connected component of the GVG. For the sake of explanation, the following discussion is limited to the GVG. However, only minor modifications are required for the higher order GVG's.

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 [1] method and Rimon's sensor based adaptation [7] use this strategy and 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 [2].

Our approach borrows some basic ideas and techniques from numerical continuation methods [5]. Continuation methods are used to 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 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 2). At x, let the hyperplane spanned by coordinates $z_2, \cdots, z_m$ be termed the "normal slice plane." We can thus decompose the local coordinates into $x = (y, \lambda)$, where $\lambda = z_1$ is termed the "sweep" coordinate and $y = (z_2, \cdots, z_m)$ are the "slice" coordinates. Now define the function $G\colon \mathbb{R}^{m-1} \times \mathbb{R} \to \mathbb{R}^{m-1}$ as follows:

$$G(y, \lambda) = \begin{bmatrix} (d_1 - d_2)(y, \lambda) \\ (d_1 - d_3)(y, \lambda) \\ \vdots \\ (d_1 - d_m)(y, \lambda) \end{bmatrix} \tag{3}$$

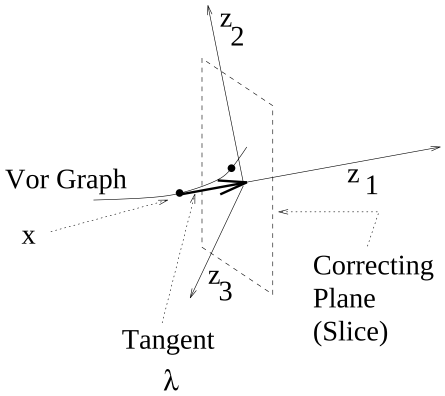

Fig. 2. Continuation Method

The function $G(y, \lambda)$ assumes a zero value only on the GVG. Hence, if 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 this function, we can locally construct an edge. While there are a number of such techniques [5], 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). 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" (Fig. 2), 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. (Fig. 2)

Let $\nabla_y G$ be the matrix formed by taking the derivative of Eq. 3 with respect to the normal slice coordinates:

$$\nabla_y G(y, \lambda) = \begin{bmatrix} \nabla_y d_1(y, \lambda) - \nabla_y d_2(y, \lambda) \\ \nabla_y d_1(y, \lambda) - \nabla_y d_3(y, \lambda) \\ \vdots \\ \nabla_y d_1(y, \lambda) - \nabla_y d_m(y, \lambda) \end{bmatrix} \tag{4}$$

where ∇_y denotes the gradient with respect to the y-coordinates. We will show that $\nabla_y G(y, \lambda)$ is full rank at $x = (y, \lambda)$, and so it is possible to use an iterative Newton's Method to implement the corrector step. If y^k and λ^k are the k^{th} estimates of y and λ, the $k + 1^{st}$ iteration is defined as

$$y^{k+1} = y^k - \left(\nabla_y G\right)^{-1} G(y^k, \lambda^k) \tag{5}$$

where $\nabla_y G$ is evaluated at (y^k, λ^k). After taking the prediction step, the goal of the correction step is to find where the GVG locally intersects the "correcting plane." (Fig. 2)

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 2D environments, or a 2×2 matrix for 3D environments).

The following demonstrates this procedure is theoretically sound and can be implemented using local information.

3.1 Properties for Tracing

Our goal in this section is to show that Eq. 5 is well defined, and that we can always compute (using local sensor data) a vector which is tangent to the GVG. The proofs of the ensuing lemmas and propositions can be found in the appendix. In proving these assertions, several new and useful properties of the Generalized Voronoi Graph are presented.

Computing the Tangent to the Graph. We first tackle the question of how to determine the tangent to a GVG edge from sensor data. Let the "Regular Voronoi Graph" (RVG) denote the Voronoi Graph for the case in which the obstacles are points. Let x be a point on the GVG edge, and let $\{c_i\}$ denote the set of closest points of the m closest obstacles $\{C_i\}$ to x. It should be noted that the RVG edge defined by the points $\{c_i\}$ and the GVG edge coincide at x. We can compute many items of interest about the GVG by exploiting the coincidence of the RVG with the GVG at x.

PROPOSITION 3.1 The tangent to a GVG edge at x is defined by the vector orthogonal to the hyperplane which contains the m closest points: $c_1, \ldots, c_m$ of the m closest objects, $C_1, \ldots, C_m$.

Proof: This proposition is a simple consequence of the following two lemmas:

LEMMA 3.2 Let $c_1, \ldots, c_m$ be the m closest obstacle points to $x \in$ RVG edge. The tangent to the RVG is orthogonal to the hyperplane containing $c_1, \ldots, c_m$.

LEMMA 3.3 Let $c_1, \ldots, c_m$ be the closest points in the m nearest obstacles to $x \in$ GVG edge. The tangent to the GVG edge at x is equal to the tangent to the RVG defined by $c_1, \ldots, c_m$.

Thus, by knowing the distance and direction to the m nearest points, the tangent to the graph is easily computed. ∎

$\nabla_y G$ is Invertible. We now show that the numerical procedure defined by Eq. 5 is well defined for $\Delta\lambda$ sufficiently small.

PROPOSITION 3.4 *(Equidistant Surface Full Rank Property)* $\nabla_y G(y, \lambda)$ has full rank (i.e., has rank $(m-1)$) in a neighborhood of the GVG on the correcting plane.

Proof: This is a simple consequence of the following two lemmas (which are proved in the appendix):

LEMMA 3.5 On the normal slice plane, $\nabla G(x)$ has rank $(m-1)$ for all $x \in \mathcal{F}^m$.

LEMMA 3.6 *$rank(\nabla_y G) = rank(\nabla G)$ for $x \in \mathcal{F}^m$.*

Since $\nabla_y G$ is an $m-1$ by $m-1$ matrix, by these lemmas, it must have rank $(m-1)$ for $x \in \mathcal{F}^m$, and therefore be invertible at x.

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

In practice, the neighborhood of invertibility is quite large with this method. Practically speaking, this result states that the numerical procedure defined by Eq. 5 will be robust for reasonable errors in robot position, sensor errors, and numerical round off.

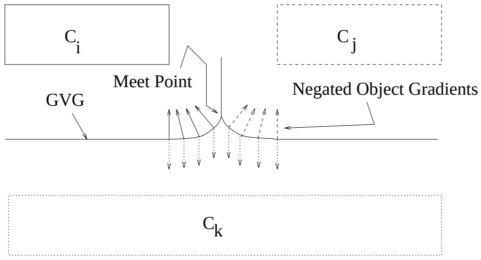

Fig. 3. Meet Point Detection

3.2 Terminating Conditions

So far, we have shown that the robot can access and trace a Generalized Voronoi Edge. Due to the boundedness of the robot's environment, the Generalized Voronoi Edges must terminate, as stated in the following proposition (whose proof is omitted due to space limitations).

PROPOSITION 3.7 Given the Equidistant Surface Transversality Assumption, a Generalized Voronoi edge must: (1) terminate at a Generalized Voronoi Vertex; (2) terminate on the boundary point of an Equidistant Surjective Surface (which includes the case of obstacle boundaries); or (3) be part of a "cycle."

We may sometime call a Generalized Voronoi Vertex a "meet point," since edges "meet" at such a vertex.

Incremental construction of the Generalized Voronoi Graph is akin to a graph search method where the Generalized Voronoi Edges are the "edges" and the meet points and boundary points are the "nodes." Once the robot has accessed a point on the GVG, it begins tracing an edge. If the robot encounters a meet point, it marks 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 hits another unvisited meet point, the above procedure is recursively repeated. When the robot hits a boundary point, it simply turns around and retraces its path to some previous meet point with unexplored directions. The robot terminates exploration of the GVG fragment (i.e., there may be other disconnected GVG fragments) when there are no more unexplored directions in any meet point. If desired, graph searching techniques such as the A-star algorithm or depth first search can be used to control the tracing procedure. A later section deals with the cycle condition.

3.3 Meet Point Detection

Finding the meet points is essential to proper construction of the graph. While a meet point occurs when the robot is equidistant to $m+1$ objects, it is unreasonable to expect that a robot can exactly detect such points. For example, while tracing an edge, it is unlikely that the robot will pass exactly through an $m+1$ equidistant point. Furthermore, sensor error may make such detection difficult. However, as shown in Fig. 3, meet points can be robustly detected by watching for an abrupt change in the direction of the (negated) gradients to the m closest obstacles. Such a change will occur in the vicinity of a meet point.

3.4 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

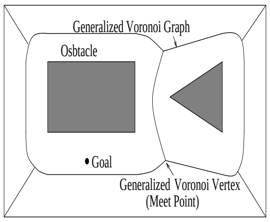

Fig. 4. Original GVG

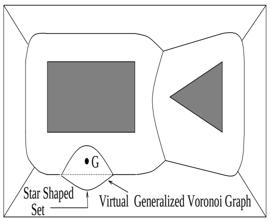

Fig. 5. Virtual GVG

$m+1$ Generalized Voronoi Edges that emanate from each meet point in order to completely construct the GVG. Note that each emanating edge corresponds to an m-wise combination of the $m+1$ closest objects. Assume that we wish to explore and trace the edge corresponding to objects $C_1, \ldots, C_m$. Proposition 3.1 yields the 1-dimensional tangent space to the Generalized Voronoi Edge corresponding to these m objects. If v is a tangent vector computed from Lemma 3.3, 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, Generalized Voronoi Edge is closer to the m objects, which define it, than any other object. This effects motion away from C_{m+1}.

4 Incremental Departability

In sensory based exploration, the robot may or may not know the goal coordinates. If the robot does not know the goal coordinates, it is assumed that the goal is defined by a beacon which the robot can detect once it is within line of sight of the beacon. We therefore would like to find a departing method in which the robot can access the goal in a straight line. Treating the goal as an object, create a "virtual" Generalized Voronoi Graph (Fig. 4). A star shaped set, bounded by the virtual GVG, surrounds the goal. A straight line path between any point on the boundary of this virtual star shaped set to the goal can be drawn. Generally, the virtual GVG is connected to the GVG and thus there is a point within line of sight of the goal on the GVG. However, as we know from the companion paper, the virtual GVG may be disconnected. In this case, it is necessary to build a link to the disconnected component that surrounds the goal. The linking strategy is a special case of the strategy one would use to link GVG cycles to other Second Order GVG edges [3].

5 Constructing the Second Order GVG

The second (and higher) order GVG can be incrementally constructed in a highly analogous fashion. The key is to define a function, G, whose roots define the Second Order GVG (the GVG^2):

$$G_2(y, \lambda) = \begin{bmatrix} (d_1 - d_2)(y, \lambda) \\ (d_3 - d_4)(y, \lambda) \\ \vdots \\ (d_3 - d_m)(y, \lambda) \end{bmatrix} \tag{6}$$

The first row of G_2 enforces equidistance between the closest objects C_1 and C_2. The remaining rows enforce equidistance between the second closest objects. The preceding algorithms and analysis apply equally well for the incremental construction of the Second Order Generalized Voronoi Edges via Eq. 6.

In summary, the GVG^2 has the same terminating conditions as the GVG: a *Second Order Meet Point*, *Second Order Boundary Point*, and a *Second Order Cycle*. The Second Order Meet Points are detected in a fashion analogous to the first order meet points—the robot looks for a change in the gradients to the second nearest object, while maintaining equidistance to the two nearest objects. Again, at a boundary, the robot simply turns around and re-traces its steps to the previous Second Order Meet Point with unexplored directions.

For incremental construction of the GVG, it can be easily shown that the m closest objects will always be within line of sight of the robot. However, occasionally there are scenarios in which the *second* closest object may not be within line of sight of the robot, thus making incremental construction the GVG^2 quite difficult. Currently, we are developing a new roadmap, termed the *Visibility Hierarchal Generalized Voronoi Graph*, constructed solely from line of sight information. However, for now we rely on an active scanning approach when the robot loses sight of its second closest obstacles.

6 Experiments

To verify the incremental construction procedure, we implemented this approach in the planar case (i.e. m = 2) on a circular mobile robot base. The mobile robot is the B12 Mobile Robot Base, produced by Real World Interface, Inc., and it is instrumented with a ring of twelve sonar sensors which provide local distance measurement information. While the sensors are quite accurate in distance measurement (on the order of 1 cm), their angular resolution is only accurate to 22^o. In terms of our algorithm, $d_i(x)$ can be accurately measured using this robot, but $\nabla d_i(x)$ will be inaccurate.

The result of one experiment is shown in Figs. 6 and 7, though many other experiments were successfully completed. In this trial, the room was "T-shaped," with the geometry of the room and the theoretical GVG shown in Fig. 6. In Fig. 7 is shown the experimental GVG constructed by the robot. The small squares denote the edge termination points, while the hatched squares represent meet points. For safety reasons, the robot does not trace the edge all the way to the wall's boundary. The octagon shown on the graph represents the scale size of the robot. The experimental GVG edges are jagged because the tangent is crudely approximated. This crude approximation in turn is a function of the angular inaccuracy of sonar distance sensors. However, the GVG is connected, and the edges are maximally far away from the workspace boundary. Note, the actual GVG construction is quite robust even with large errors in distance measurements. A three-dimensional simulation is underway.

7 Conclusion

This paper introduced an incremental procedure to construct the GVG and the HGVG. This procedure requires only local sensor distance measurement data, and is therefore practically implementable, as demonstrated by our experiments. Hence, the Generalized Voronoi Graph and Hierarchical Generalized Voronoi Graph introduced in [4] appear to be useful means for implementing sensor based motion planning algorithms. We also believe that with small modifications, some of the numerical methods introduced in this paper can also be useful for "sensorizing" other (e.g., the OPP method) robot motion planners.

References

[1] J.F. Canny and M.C. Lin. An Opportunistic Global Path Planner.

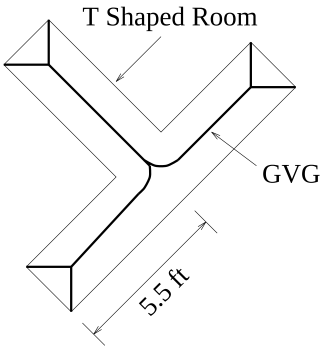

Fig. 6. Room with Actual GVG

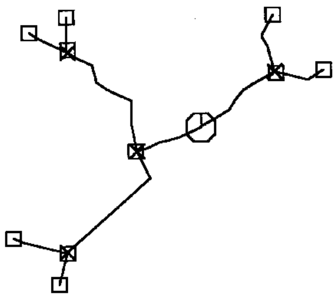

Fig. 7. Experimental GVG

Algorithmica, 10:102–120, 1993.

[2] 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.

[3] H. Choset and J.W. Burdick. Sensor Based Planning: Details of the Generalized Voronoi Graph. Technical report, Caltech, Pasadena, CA, February 1995. Available via email: choset@robby.caltech.edu.

[4] H. Choset and J.W. Burdick. Sensor Based Planning, Part I: The Generalized Voronoi Graph. In *Submitted to Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, 1995.

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

[6] N. Rao, N.S.V. 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.

[7] 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.

[8] Rao, N.S.V. Kareti, S. Shi, W. and Iyenagar, S.S. Robot Navigation in Unknown Terrains: Introductory Survey of Non-Heuristic Algorithms. *Oak Ridge National Laboratory Technical Report*, ORNL/TM-12410:1–58, July 1993.

Appendix

Proof of Lemma 3.2.

Proof: Recall that the edges of the RVG are straight line segments. So, the tangent to a point x on an edge coincides with the edge. Let the hyperplane which contains $\{c_1, \ldots, c_m\}$ be called the "base plane," and embedded in the base plane, there is an $m-2$ dimensional sphere, S, defined by $\{c_1, \ldots, c_m\}$. Finally, there exists a cone whose vertex is x and whose base is S (Fig. 8). The centerline of this cone coincides with the RVG edge and the tangent at x on this edge. Clearly the base of this cone is orthogonal to the centerline. Due to space limitations, the proof for this is omitted. Since, by definition the tangent to the RVG is orthogonal to the normal slice plane and the normal slice plane and base plane are parallel, the tangent is orthogonal to the base plane, the plane which contains the m nearest points. ∎

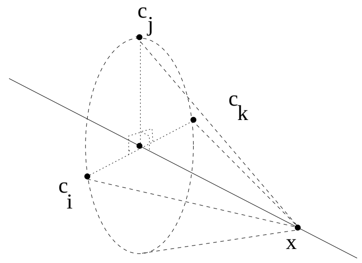

Fig. 8. Cone formed by points

Proof of Lemma 3.3.

Proof: A GVG edge can be defined by $G^{-1}(0)$ where

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

The tangent space at a point $x \in G^{-1}(0)$ is simply the null space of $\nabla G(x)$. Let $\{c_i\}$ denote the closest points to x in the m closest obstacles. The Regular Voronoi Graph for the set of points $\{c_i\}$ is defined as:

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

Since the set of closest points, $\{c_i\}$, is the same for the GVG and RVG at x, $G(x) = VG(x)$ at x. Furthermore, $\nabla G(x) = \nabla VG(x)$ at x, and thus they have the same null spaces. Hence, the GVG and RVG have the same tangent space. ∎

Proof of Lemma 3.5.

Proof: The respective tangent spaces of $\mathcal{S}_{ij}$ and $\mathcal{S}_{ik}$ are:

$$\begin{aligned} T_x\mathcal{S}_{ij} &= \{v \in T_x\mathbb{R}^m : \nabla(d_i - d_j)(v) = 0\} \\ T_x\mathcal{S}_{ik} &= \{v \in T_x\mathbb{R}^m : \nabla(d_i - d_k)(v) = 0\} \end{aligned}$$

$\nabla(d_i - d_k)$ should be interpreted a co-vector. Thus, the dot product of $\nabla(d_i - d_k)$ and v is written as $\nabla(d_i - d_k)(v)$.

By the Equidistant Surface Transversality Assumption in [4] we know that $\mathcal{S}_{ij} \pitchfork \mathcal{S}_{ik}$. Assume at some point x, $\nabla(d_i - d_j) = \kappa\nabla(d_i - d_k)$. By definition, for all $w \in T_x\mathcal{S}_{ij}$, $\nabla(d_i - d_j)(w) = 0$. Since $\nabla(d_i - d_j) = \kappa\nabla(d_i - d_k)$, for $w \in T_x\mathcal{S}_{ij}$, $\nabla(d_i - d_k)(w) = 0$. This implies that $T_x\mathcal{S}_{ij} = T_x\mathcal{S}_{ik}$ which violates the Equidistant Surface Transversality Assumption [4]. Therefore, $\nabla(d_i - d_j)(w) \neq \kappa\nabla(d_i - d_k)(w)$, that is they are linearly independent.

So for three objects: $\nabla d_i, \nabla d_j$, and ∇d_k

$$rank \begin{bmatrix} \nabla d_i - \nabla d_j \\ \nabla d_i - \nabla d_k \end{bmatrix} = 2$$

For the case of 4 nearest objects

$$\nabla G = \begin{bmatrix} \nabla d_i - \nabla d_j \\ \nabla d_i - \nabla d_k \\ \nabla d_i - \nabla d_l \end{bmatrix}$$

The Equidistant Surface Transversality Assumption [4] guarantees each row is pairwise linearly independent.

$$\begin{aligned} \nabla d_i - \nabla d_j &\neq \kappa_{jk}\big(\nabla d_i - \nabla d_k\big) \\ \nabla d_i - \nabla d_k &\neq \kappa_{kl}\big(\nabla d_i - \nabla d_l\big) \\ \nabla d_i - \nabla d_l &\neq \kappa_{jl}\big(\nabla d_i - \nabla d_j\big) \end{aligned} \tag{7}$$

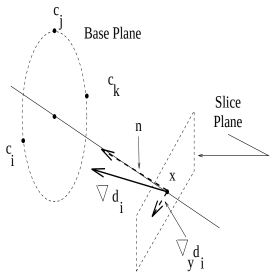

Fig. 9. n defined

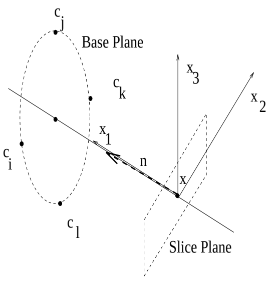

Fig. 10. Coordinate Frame

It remains to show that no one row is a linear combination of the other two. Again, we will prove this by contradiction. Assume $\nabla(d_i - d_j) = \alpha(\nabla(d_i - d_k)) + \beta(\nabla(d_i - d_l))$. Again by definition, for all $w \in T_x \mathcal{S}_{ij}$, $\nabla(d_i - d_j)(w) = 0$. Thus,

$$\begin{aligned} &\nabla(d_i - d_j) = \alpha(\nabla(d_i - d_k)) + \beta(\nabla(d_i - d_l)) \\ &\Rightarrow \Big(\alpha\big(\nabla(d_i - d_k)\big) + \beta\big(\nabla(d_i - d_l)\big)\Big)(w) = 0 \\ &\Rightarrow \Big(\big(\nabla(d_i - d_k)\big) + \tfrac{\beta}{\alpha}\big(\nabla(d_i - d_l)\big)\Big)(w) = 0 \end{aligned}$$

Since by Equidistant Surface Transversality Assumption [4], for all $w \in T_x S_{ij}$:

$$\begin{aligned} \big(\nabla(d_i - d_k)\big)(w) &\neq 0 \\ \big(\nabla(d_i - d_l)\big)(w) &\neq 0 \end{aligned}$$

this implies that $\nabla(d_i - d_k) = \frac{\beta}{\alpha}\big(\nabla(d_i - d_l)\big)$. However, this contradicts one of the three inequalities in equation 7. Therefore, all the rows of W are linearly independent of each other and $rank(W) = 3$. The Lemma follows by induction. ■

Proof of Lemma 3.6.

Proof: First, we must consider the following lemmas. The proof for Lemma A.1 is given below.

LEMMA A.1 Let x be a point in the GVG. Let $C_1, \cdots, C_m$ be the m closest objects to x. The projections of ∇d_i, $i = 1, \cdots, m$ onto the normal slice at x have the same length.

$$|\pi_y \nabla d_j| = |\pi_y \nabla d_k| \,\forall\, j, k$$

COROLLARY A.2 In the slice coordinates, the first coordinate of the single object distance gradients (for the m nearest obstacles) have the same value.

$$\nabla d_i = \begin{bmatrix} \kappa & \nabla_y d_i \end{bmatrix}$$

Proof: Let n be the tangent vector to the Generalized Voronoi Graph, such that $n = \nabla d_i - \nabla_y d_i$. So, $\langle n, \nabla_y d_i \rangle = 0$ (Fig. 9). In the proof of Lemma A.1), it was shown that the angles between n and any $\nabla_y d_i$, are all the same. Define a coordinate system with origin at x and basis vector x_1 pointing along n. x_1 is normal to the normal slice plane (Fig. 10). The gradients all have the same 1^{st} coordinate, κ where $\kappa = \|n\|$. ▼

Using these results, the proof of Lemma 3.6 follows easily. By Corollary A.2, the first component of ∇d_i is κ for all i, and the last $m-1$ coordinates are $\nabla_y d_i$. Since the $i - 1^{st}$ row of ∇G is of the form $\nabla d_1 - \nabla d_i$, the first column of ∇G is zero for x in the GVG:

$$\begin{aligned} \nabla_y d_1 - \nabla_y d_i \;&\rightarrow \begin{bmatrix} \kappa & \nabla_y d_1 \end{bmatrix} - \begin{bmatrix} \kappa & \nabla_y d_i \end{bmatrix} \\ &= \begin{bmatrix} 0 & \nabla_y d_1 - \nabla_y d_i \end{bmatrix} \end{aligned}$$

So, $\nabla G = \begin{bmatrix} 0 & \nabla_y G \end{bmatrix}$. Thus, $rank(\nabla G) = rank(\nabla_y G)$. ■

Proof of Lemma A.1.

Proof: First, the Lemma is proven for the RVG, and then is extended to the GVG. For the RVG, our objects are just m points, $\{c_1, \ldots, c_m\}$. We rely on the following Lemma, whose proof is omitted.

LEMMA A.3 The normal slice plane and the correcting slice plane are parallel to the plane which contains the m nearest points, for both the GVG and RVG.

LEMMA A.4 On a hyperplane orthogonal (normal slice plane) to a tangent vector of an RVG edge at a point, x, $\forall c_j, c_k$, $j, k \in \{1, \ldots, m\}$ (m closest equidistant points),

$$|\pi_y \nabla d_j| = |\pi_y \nabla d_k|$$

Proof: Recall from Lemma 3.2 that the tangent to the RVG is the vector orthogonal to the hyperplane, termed the base plane, that contains the m nearest points, $c_1, c_2, \ldots, c_m$. We will show that the lengths of all of the gradients projected onto the base plane are the same, and since by Lemma A.3, the base plane and slice plane are parallel, the lengths of the gradients projected onto the slice plane are the same.

We first show the lengths of the projected gradients onto the base plane are the same by showing the angles between the normal of the base plane and all of the gradietns are the same. The m closest points define an $m-2$ dimensional sphere, S, which is contained in the base plane. Define a coordinate system with origin at x and with x_1 axis collinear with the RVG edge, which passes through the center S. Let n be a unit vector pointing in the x_1 direction. See Fig. 9.

For any q_i and $q_j \in S$, we need to show that the angle, between $\overline{xq_i}$ and n (i.e., the x_1 axis), and the angle between $\overline{xq_j}$ and n are the same. Let q_i and q_j be the vectors $\overline{xq_i}$ and $\overline{xq_j}$, respectively.

$$\frac{\langle n, q_i \rangle}{\|n\|\|q_i\|} = \frac{\langle n, q_j \rangle}{\|n\|\|q_j\|}$$

This is equivalent to showing that $\langle n, q_i \rangle = \langle n, q_j \rangle$ because $\|q_i\| = \|q_j\|$. Let the distance between the slice plane and the base plane be a, thus, the first coordinate of q_i and q_j is a.

$$\begin{aligned} \langle n, q_i \rangle &= \langle (1, 0, \ldots, 0)^T, (a, q_i^2, \ldots, q_i^m)^T \rangle &= a \\ \langle n, q_j \rangle &= \langle (1, 0, \ldots, 0)^T, (a, q_j^2, \ldots, q_j^m)^T \rangle &= a \end{aligned}$$

Recall, n is collinear with the RVG edge, and thus the angle between any point on the sphere S, and the RVG edge is the same. This is equivalent to saying that the angle between any gradient at x and the RVG edge is the same because one can define the gradient as $\nabla d_i = \frac{q_i}{\|q_i\|}$ Thus, since the lengths of all the gradients are the same, the lengths of all the projected gradients onto the base plane are the same. Furthermore, since the base plane and normal plane are parallel (i.e., they have the same normal), the lengths of all the projected gradients on the normal slice plane are the same. See Fig. 9. ▼

In the more general case, a cone is defined by the set of m points $\{c_1, \ldots, c_m\}$, where c_i is the closest point on C_i to x. Furthermore, an RVG edge is also defined by these m points. Again by Lemma 3.3 the GVG and RVG tangents are equal, so the GVG and RVG normal slice plane are also the same, and by assumption, both orthogonal to the tangent. The gradient vectors emanating from x all have the same angle with respect to the slice plane's normal. Thus, the lengths of the gradients projected onto the slice plane are equal. ■

Acknowledgements: The authors gratefully acknowledge the support of the Office of Naval Research, Grant # N00014-93-1-0782. The authors also would like to thank Richard Murray, Jim Ostrowski, Andrew Lewis, Dave Kriegman, Luis Goncalves, Jonah Harley, and Elon Rimon for their input into the work found in both this paper and the companion paper.