Mobile Robot Navigation: Implementing the GVG in the Presence of Sharp Corners

Ilhan Konukseven** and Howie Choset*

Abstract *A robot can explore an unknown environment by incrementally constructing a roadmap of that environment using line of sight sensor information. Recall that a roadmap is a one-dimensional representation of a robot's environment. This paper addresses one problem that occurs while generating a roadmap: what happens when sonar sensors cannot detect sharp objects because of the specularities? To do this, a new sensor model is combined with an already existing incremental construction procedure for a roadmap. Experiments on a mobile robot validate the results of this paper.*

1 Introduction

Sensor based planning integrates sensor information into the planning process, in contrast to *classical planning*, which requires that full knowledge of the world be available to the robot prior to the planning event. We believe that the realistic deployment of robots into unknown environments requires sensor based planning. Similarly, when full knowledge of the environment is available, but is too difficult to input into the robot, sensor based planning bypasses the need to enter the environmental model: the robot simply explores the environment and builds up its own representation.

This work uses a sensor based planning approach that relies on a *roadmap*, a representation which captures all of the salient geometric features found in a robot's free space. Once a robot constructs a roadmap, it has in effect, explored an environment. Previous work [5] includes the prescription of an incremental construction procedure for a particular class of roadmaps. This procedure requires only line of sight information, which is an upper bound to what sensors provide, and is thus amenable to sensor based implementation.

Unfortunately, this approach does not consider issues, such as sensor quantization and the physics of sonar sensors. The long-term goal of this work is to adapt the incremental construction procedure for realistic sensor based use. This paper presents one aspect of this work: how to deal with objects in an environment with sharp corners that the sonar sensors cannot detect? The approach here exploits the fact that the robot is generating a roadmap and thus handle situations where objects become "invisible" because of sharp corners. (Note that the methods in this paper generalize to higher dimensions, and thus a more broad class of robots, other than mobile robots.)

2 Relation to Previous Work

Much of the previous work in sensor based planning is not complete and is limited to the plane. One class of heuristic algorithms employs a behavioral based approach in which the robot is armed with a simple set of behaviors (e.g., following a wall) [2]. Another heuristic approach involves discretizing a planar world into pixels of some resolution. Typically, this approach handles errors in sonar sensing readings quite well by assigning each pixel a value indicating the likelihood that it overlaps an obstacle [1]. Strong experimental results indicate the utility of these approaches, and thus these algorithms may provide a future basis for complete sensor based planners. Unfortunately, these approaches neither afford proofs of correctness that guarantee a path can be found, nor offer well established thresholds for when these heuristic algorithms fail. Finally, these approaches do not typically generalize into higher dimensions.

There are many non-heuristic algorithms for which provably correct solutions exist in the plane (see [9] for an overview). Our approach is to adapt the structure of a provably correct classical motion planning scheme to a sensor based implementation. One such approach is based on a roadmap [3], a one-dimensional subset of a robot's free space which captures all of its important topological properties. A roadmap has the following properties: *accessibility*, *connectivity*, and *departability*. 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 path onto the roadmap (accessibility), traversing the roadmap to the vicinity of the goal (connectivity), and then constructing a path from the roadmap to the goal (departability). These methods are useful in higher dimensions because the bulk of the motion planning is done in a one-dimensional space.

We chose roadmaps because of their concise representation and their upward compatibility into higher dimensions. Roadmaps are useful in m-dimensional spaces because a bulk of motion planning occurs on the one-dimensional roadmap. Roadmaps are also concise in that they do not require that the entire environment be discretized into a fine resolution of pixels.

The roadmap, used in this work, can trace its roots to the *generalized Voronoi diagram* (GVD) in the plane

*Carnegie Mellon University **CMU, METU (NATO Science Fellowship Program by TUBITAK)

(i.e., when $m = 2$). Ó'Dúnlaing and Yap [8] first applied the GVD, which is the locus of points equidistant to two or more obstacles, to motion planning for a disk in the plane. However, the method in [8] requires full knowledge of the world's geometry prior to the planning event. In [10], an incremental approach to create a Voronoi Diagram-like structure, which is limited to the case of a plane, was introduced.

The GVG's incremental construction procedure [5] gives the GVG its primary strength. This incremental construction procedure only requires line of sight information and this procedure places no restrictions on the type of obstacles; obstacles need not be polygonal, polyhedral, nor convex, which are assumptions most motion planners require.

For some environments, this algorithm has been successfully implemented on a mobile robot with a ring of sonar sensors [6]. Unfortunately, the incremental construction procedure does not take into consideration the properties of the sonar sensors such as specularities, sensor range, and quantization. For example, sonar sensors are not good at detecting sharp corners that may appear in typical environments. A sharp corner is "invisible" to the sensor that is facing it. The paper deals with that situation (note the sharp corner problem for GVG edge tracing was first pointed out in [6].

3 Related Work

The work presented in this paper is based on the GVG, which is described in [4], [5], [6]. A review of the GVG and its incremental construction procedure is included below for the sake of completeness, but it could be omitted by a reader already familiar with this work.

3.1 Distance Function

Assume the robot is a point operating in a work space, $\mathcal{W}$, which is a subset of an m-dimensional Euclidean space, $\mathbb{R}^m$. $\mathcal{W}$ is populated by convex obstacles $C_1, \ldots, C_n$. Non-convex obstacles are modeled as the union of convex shapes. The distance between a point and an obstacle is the shortest distance between the point and all points in the obstacle. The distance function, and its "gradient," respectively are

$$d_i(x) = \min_{c_0 \in C_i} \|x - c_0\| \quad \text{and} \quad \nabla d_i(x) = \frac{x - c_0}{\|x - c_0\|},$$

where (1) d_i is the distance to obstacle C_i from a point x, and (2) the vector $\nabla d_i(x)$ is a unit vector in the direction from x to c_0, where c_0 is the nearest point to x in C_i. Typically, the environment contains multiple obstacles, and thus distance is measured to multiple obstacles with the multi-object distance function, $D(x) = \min_i d_i(x)$

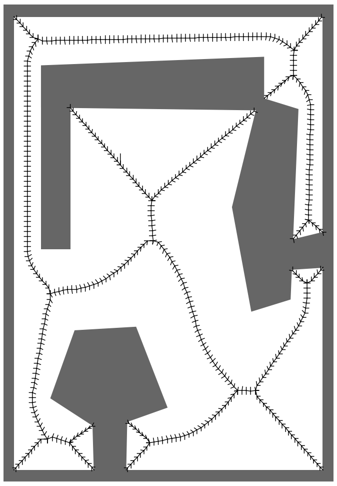

Fig. 1. The ticked line segments are the planar GVG for the bounded environment. The ticks point at the nearest point on an obstacle, and are thus the negated gradients.

3.2 The Generalized Voronoi Graph

The basic building block of the GVG is the set of points equidistant to two sets C_i and C_j, such that each point in this set is closer to the objects C_i and C_j than any other object. We term this structure the *two-equidistant face*,

$$\mathcal{F}_{ij} = \{x \in \mathbb{R}^m : 0 \le d_i(x) = d_j(x) \le d_h(x) \quad \forall h \ne i, j \text{ and } \nabla d_i(x) \ne \nabla d_j(x)\}.$$

A two-equidistant face has co-dimension one in the ambient space, and thus in the plane, a two-equidistant face is one dimensional [4].

The Pre-image Theorem asserts that the union of the two-equidistant faces, i.e., the GVD, is $(m-1)$-dimensional [4]. 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 $\mathcal{F}_{ijk}$. That is,

$$\mathcal{F}_{ijk} = \mathcal{F}_{ij} \bigcap \mathcal{F}_{ik} \bigcap \mathcal{F}_{jk}$$

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.)[4]

The *generalized Voronoi graph* (GVG) is the collection of m-equidistant faces and $m+1$-equidistant faces. Later, the m-equidistant faces are termed *generalized Voronoi 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. In the planar case, the GVG and GVD coincide (See Fig. 1).

3.3 Incremental Construction of the GVG

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

Incremental Accessibility. The robot accesses the GVG by increasing its distance to the nearest obstacle. Then, while maintaining double equidistance, the robot increases its distance from the two closest obstacles until it is three-way equidistant. This procedure is repeated, until the robot is m-wise equidistant. In the planar case, the robot simply moves in a direction opposite to which the nearest sensor is

Tracability. Once the robot accesses the GVG, it must trace it out. In an incremental context, the property of connectivity is interpreted as *tracability.* More specifically, tracability implies that using only local data, the robot can: (1) "trace" the GVG edges; (2) determine when to terminate the edge tracing process, and (3) determine when to start new edge tracing procedures.

The GVG incremental approach to edge construction borrows ideas from numerical continuation methods [7]. Continuation methods 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

$$G(y,\lambda) = \begin{bmatrix} (d_1 - d_2)(y,\lambda) \\ (d_1 - d_3)(y,\lambda) \\ \vdots \\ (d_1 - d_m)(y,\lambda) \end{bmatrix} \qquad (1)$$

Since the G is zero on the GVG, tracing the roots of G is akin to generating a GVG edge. Note that G comprises the distance function and thus the GVG can be generated from sensor data. See [5] for details of the edge tracing technique.

Terminating Conditions. The explicit terminating conditions for edge tracing are described in [5], but in the planar case there are two terminating conditions:

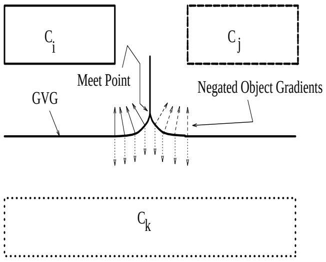

Fig. 2. Meet Point Detection.

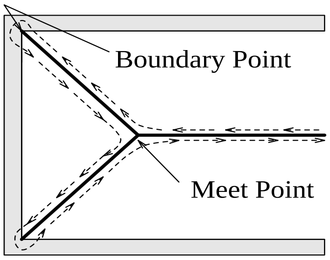

Fig. 3. Boundary Point. The arrows delineate the path a robot would follow while construction the GVG.

a meet point, where three GVG edges join, and a boundary point, where a GVG edge intersects the boundary of the environment.

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 2, 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.

After reaching a meet point, the robot explores an unexplored edges emanating from the meet point. At a boundary point, the robot simply back tracks to a previous meet point that has unexplored GVG edges associated with it. See Figure 3. Once all meets points have no unexplored edges emanating from them, the GVG has been fully explored. Note that incremental construction of the GVG is akin to a graph search where GVG edges are the "edges" and the meet points and boundary points are the "nodes." 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.

4 Previous Implementation

The GVG scheme was implemented on a mobile robot with a ring of ultrasonic sensors radially distributed around the perimeter of the robot. These sensors determine distance by measuring the time of flight of the ultrasound pulses that reflect off an object and return to the sensor. Although these sensors provide accurate distance measurements, their readings are not precise in the azimuth.

The GVG incremental construction procedure has been implemented using a simplistic sensor model [6]. This model assumes that the sensors are rigidly at-

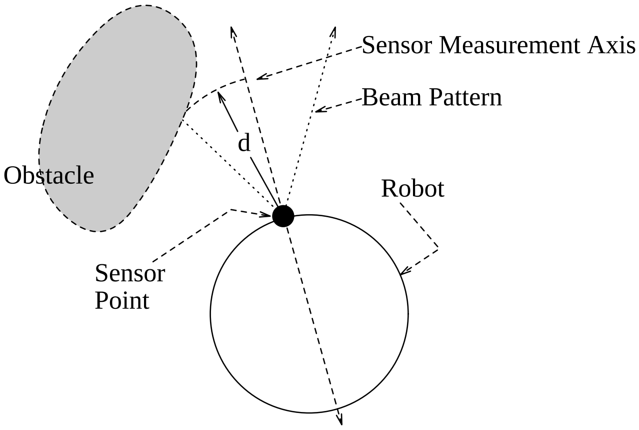

Fig. 4. Simplified Distance Measurement Sensor Model

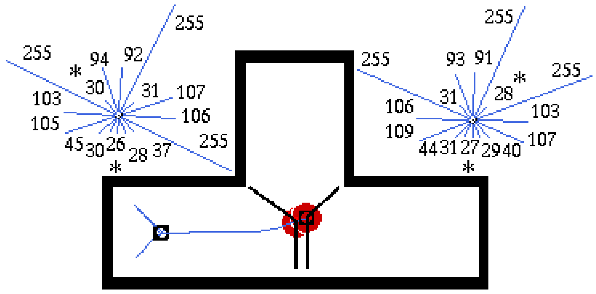

Fig. 5. Naive Meet Point Detection (Later, this will be an "event" detection).)

tached, pointing radially outward from the robot. 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 Fig. 4). Local minima of the distance readings correspond to distance to the nearby obstacles. [6].

Recall that a meet point $\mathcal{F}_{123}$ the robot terminates the edge tracing procedure of a GVG edge $\mathcal{F}_{12}$, and begins tracing edge $\mathcal{F}_{13}$. In order to do this, the robot first must determine when it encounters a meet point. When there is an "abrupt" change in a sensor associated with one of the two closest obstacles, then the robot has passed by a meet point. Since the robot has few sensors and thus a low resolution, an "abrupt" change is indicated by a shift of the local minimum by more than one sensor location (Fig. 5).

Our experiments indicate that this sensor model and meet point detection scheme work well when the two closest features on the two closest obstacles are flat faces, curved faces, and gentle corners. This is so, because the sensors with the two smallest local minima (i.e., those that correspond to the two closest obstacles) have a sensor centerline axis that is nearly parallel to the obstacle normals. See Figure 6 for the results of an experiment where the robot generates a GVG of an unknown environment.

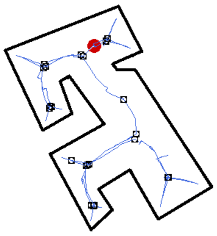

Fig. 6. Experiment.

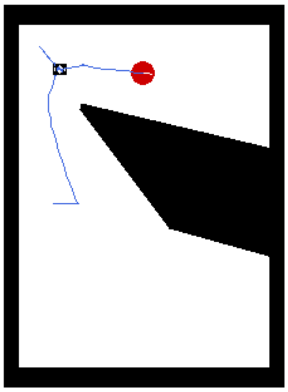

Fig. 7. Working.

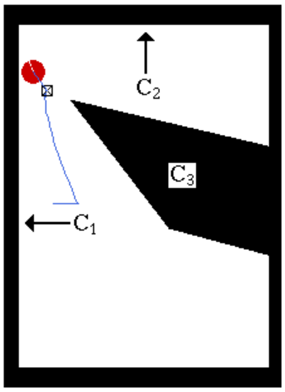

Fig. 8. Not Working

5 Problem with Previous Sensor Model

Unfortunately, the previous sensor model gives false meet points and does not provide an accurate GVG representation. Note the differences between the environments in Figs. 7 and 8. In Fig. 7, C_3 has a blunt protrusion (not sharp) in contrast to C_3 in Fig. 8. Now, compare the GVG traces in Figs. 7 and 8. The GVG trace in Fig. 7 is correct, whereas the one in Fig. 8 is not because there is an incorrect meet point.

In both cases, a meet point is detected when there is an abrupt change in one of the two closest obstacles, as perceived by the sensors. However in Fig. 8, there is an abrupt change in one of the two closest obstacles because the distance to C_3 became infinite as a result of the sonar sensor's inability to see C_3 because of its sharp corner. In effect, C_3 became "invisible" to the robot and the two perceived closest obstacles become C_1 and C_2. See Figs. 9 and 10. Note that the naive sensor model would consider C_3 to be visible.

6 Sharp Corners

Note that the problem of "false" meet points was first pointed out in [6]. In that paper, the authors felt that there was a problem with the meet point detection scheme. In actuality, the problem lies with the sensor model. Now, we introduce a new sensor model and enumerate all the conditions in which a sharp object can cause problems. Next, we eliminate most of the conditions, and show that only three such conditions can

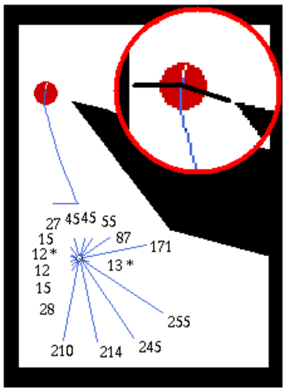

Fig. 9. Robot has traced out GVG edge fragment. The lower left portion displays the sonar sensor values and the upper-right portion contains a closeup of the robot and its two local minima.

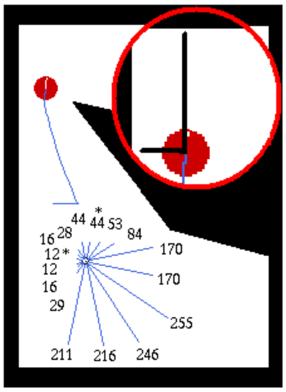

Fig. 10. Robot's sensors cannot see C_3 and thus C_1 and C_2 become to two closest obstacles.

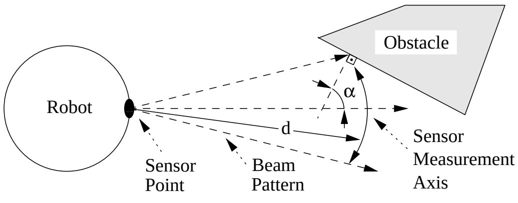

Fig. 11. New Sensor Model.

occur and provide solutions for them.

The sensor model is an adaption of the previous one. Again, the sensors measure distance to nearby obstacles, along a fixed direction termed the *sensor measurement axis*. However, if the normal to the obstacle and the sensor measurement axis form an angle greater than α, then the sensor does not detect the obstacle, although it is in the sensor cone (Figure 11) Therefore, this model takes into consideration the specularities of the sonar sensors because when the normal and sensor measurement axis form an angle greater than α, the ultrasound hits the obstacle, but does not bound back to the sensor.

So, while the robot is tracing the edge in Fig. 8, the new sensor model indicates that the distance to C_3 becomes infinite because the normals of C_3 form an angle greater than some pre-specified α. The naive meet point detection scheme would indicate that this is a meet point, when in fact it is not. However, such a situation is termed an *event.*

Definition 6.1 (Event) *An* **event** *occurs when a robot is initially tracing a GVG edge associated with objects C_i and C_j, and then sensors perceive a third object C_k that is* not farther *to the robot than either C_i or C_j.*

First, consider all possible relationships among d_i, d_j, and d_k as the robot *continuously* moves along the GVG. That is,

$$d_{i,j,k}\left\{\begin{array}{c}<\\=\\>\end{array}\right\}d_{j,k,i}\left\{\begin{array}{c}<\\=\\>\end{array}\right\}d_{k,i,j}\left\{\begin{array}{c}<\\=\end{array}\right\}\infty$$

After enumerating all possibilities above, and reducing all similar expressions (recall that $d_i = d_j$ prior to the event, so they can be interchanged), following relationships result.

Case	Dist. Rel. (a)	Dist Rel. (b)
1.	$d_i = d_j = d_k < \infty$	$d_i = d_j = d_k = \infty$
2.	$d_i = d_j < d_k < \infty$	$d_i = d_j < d_k = \infty$
3.	$d_i = d_k < d_j < \infty$	$d_i = d_k < d_j = \infty$
4.	$d_i < d_j = d_k < \infty$	$d_i < d_j = d_k = \infty$
5.	$d_k < d_i = d_j < \infty$	$d_k < d_i = d_j = \infty$
6.	$d_i < d_j < d_k < \infty$	$d_i < d_j < d_k = \infty$
7.	$d_i < d_k < d_j < \infty$	$d_i < d_k < d_j = \infty$
8.	$d_k < d_i < d_j < \infty$	$d_k < d_i < d_j = \infty$

Note that continuity of the distance function in a bounded environment precludes the possibilities of conditions 1b, 3a, 4a, 4b, 6a, 6b, 7a, 8a, and 8b. The intuition behind the continuity argument can be explained with condition 8b. Prior to the event, $d_i = d_j < d_k$. Then, at the the event the robot perceives C_k as the closest obstacle, C_i as the second closest obstacles, and C_j as the third closest obstacle. Note that C_j is "invisible" to the robot and C_i is not (and thus the distance to C_i is finite). In order for d_i to be greater than d_k, the robot must pass through a point where $d_i = d_k$, by continuity of the distance function. Condition 8b indicates that the robot did not pass through such a point and that the distance function is not continuous. Since the distance function is continuous, condition 8b cannot occur (nor be an event).

Also note, conditions 2a and 2b are not events, but rather the appropriate distance relationship the robot perceives while tracing a GVG edge. Therefore, the following conditions are events.

Case	Distance Relation	Condition
1a.	$d_i = d_j = d_k < \infty$	meet point
3b.	$d_i = d_k < d_j = \infty$	sharp-meet point
5a.	$d_k < d_i = d_j < \infty$	post-meet point
5b.	$d_k < d_i = d_j = \infty$	two sharp objects
7b.	$d_i < d_k < d_j = \infty$	pre-meet point

Condition 1a is the event that corresponds to a meet point because the robot is triply equidistant to three obstacles. Continuity of the distance function guarantees that condition 3b is also a meet point, but one of the two closest obstacles became invisible when the meet point was detected. This is a rare condition, and thus we did not implement it in our mobile robot experiments.

Now, let's consider condition 7b because it appears in the example described in the previous section. Note

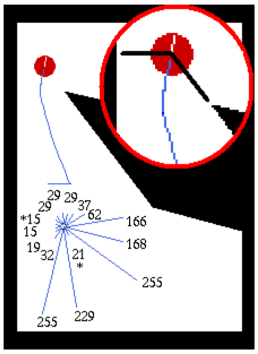

Fig. 12. Robot infers (does not sense) distance to one of the closest obstacles.

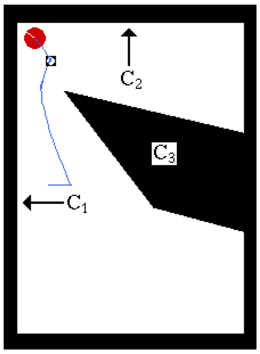

Fig. 13. Robot detects the correct meet point.

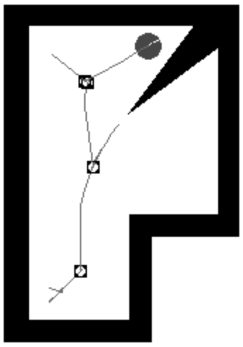

Fig. 14. Robot infers (does not sense) distance to next closest obstacle.

that in Fig. 10 the values of the two smallest local minima (distance to two smallest objects) are not the same, nor close to each other. That is, $d_1 < d_2 < d_3 = \infty$ which corresponds to condition 7b because the robot was initially tracing edge $\mathcal{F}_{13}$.

With the new sensor model, sharp corners are the only features which can become "invisible," as what transpires in condition 7b. When this occurs, using the previous distance measurements, the robot can determine the location, in world coordinates, of the sharp corner and continue measuring distance to the sharp corner as the GVG edge is traced. Recall, that the GVG edge tracing technique requires only the distance to the two closest obstacles to the robot. Now, that information is still available to the robot, although the robot cannot "see" one of the two closest obstacles.

So, when the robot reaches the configuration in Fig. 10, it determines the location of the sharp corner and continues tracing the edge until the obstacle is visible again (Figs. 12). In 13, the robot finds the correct meet point. Note that condition 5b is identical to condition 7b, except the robot determines the location of two sharp corners, instead of one.

In condition 5a, the robot traced by a meet point because it could not "see" the next closest object. In this condition, the robot may need to touch the environment. In Fig. 14, the robot traced by the meet point because the next closest obstacle had a sharp corner and was invisible to the robot's sonar sensors. The robot terminates the edge tracing procedure when its bumper sensor is activated, i.e., $0 = d_k < d_i = d_j$. Again, the location of the sharp corner is determined and the robot traces back along double equidistance between C_i and C_j until $d_i = d_j = d_k$. Knowing the location of the sharp corner, the robot can determine all of the GVG edges that emanate from the meet point, and begin new edge tracing processes. Just like before, all of this can occur without the robot explicitly "seeing" the obstacles.

7 Conclusion

The GVG incremental construction procedure uses a new sensor model, introduced in this paper. This new sensor model takes into consideration specularities of sonar sensor data. Future work includes an experimental determination of the α parameter in this sensor model and a rigorous derivation to determine the minimum of sensors a mobile robot must have to navigate an unknown environment.

References

[1] J. Borenstein and J. Koren. Real-time Onstacle Avoidance for Fast Mobile Robots in Cluttered Environments. In *IEEE Conference of Robotics and Automation*, pages 572–577, Cincinnati, Ohio, May 1990.

[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. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA, 1988.

[4] 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.

[5] 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.

[6] H. Choset, I. Konuksven, and J.W. Burdick. Sensor Based Planning for a Planar Rod Robot. In *Proc. IEEE/SICE/RSJ Int. Conf. on Multisensor Fusion on Multisensor Fusion and Integration for Intelligent Systems*, Washington, DC, 1996.

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

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

[9] 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.

[10] 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.