Toward Robust Sensor Based Exploration by Constructing Reduced Generalized Voronoi Graph

Keiji Nagatani

Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Howie Choset

Mechanical Engineering and
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Abstract

Our research considers sensor based mobile robot exploration in an unknown indoor environment. One of the big problems in this area is positioning error. To deal with this problem in the exploration task, we had proposed a localization method using a partially constructed topological map termed the generalized Voronoi graph (GVG). The GVG-based method performed successfully in some large scale indoor environments, but had problems resulting from poor sensing ability of the sixteen sonar sensors on-board the robot. The poor sensing ability was particularly deleterious at geometric structures in the GVG that were "unstable." To solve the problem, we adopted "Reduced Generalized Voronoi Graph (RGVG)" for exploration map instead of original GVG. The RGVG is a stable subset of the GVG and is sufficient for motion planning. To enable realistic and reliable exploration using RGVG structure, we developed an edge matching procedure for topology matching. In this paper, we introduce the RGVG structure and the edge matching procedure for robust exploration. Experimental results verify the described work.

1 Introduction

Sensor based exploration is a fundamental function for mobile robot intelligence. We adopted the incremental construction procedure of the Generalized Voronoi Graph (GVG) [1] for our exploration strategy. In two dimensional case, the GVG is the set of points equidistant to two objects. Therefore, exploration is executed by tracing the set of points equidistant to two objects until it reaches a *meet point*, a point equidistant to three objects, or a *boundary point*, a point where the GVG intersects the boundary. Many GVG edges terminate at a meet point, so the robot picks an edge and traces it until the robot encounters another meet point, where this procedure is repeated again. If the robot arrives at a boundary point it backtracks to a meet point with an unexplored direction, and pursues a new edge. When all meet points have no unexplored directions, exploration is done.

The original implementation only worked in small-sized environment using a conventional wheeled mobile robot (Total run was less than about 10 meters) [2]. It was not suitable for large-scale environments because of accumulated positioning error in the dead reckoning system.

To solve the problem, we had developed a topological matching technique for localization [3]. In this approach, localization is done by matching meet points. If each meet point was unique, then the robot need only to look at its sensors to determine where it is in the map. Many times, meet points "look" the same, so the robot must consider the current meet point and its neighbors. Essentially, the robot is performing a graph matching technique with a partially constructed GVG. This is why this method is called topological matching.

This approach worked in some large scale environment, however it still required a highly structured environment – sometime we used cardboard to enforce structure. Once we started experimenting in less-structured environment, the *weak meet point problem* appeared. If the environment is slightly perturbed, a weak meet point could appear or disappear, whereas strong meet points are constant. Although the environments we consider are static, sensor error can give the robot the illusion that the environment was slightly perturbed. This presents a problem when performing localization based on meet points.

To deal with this problem, we defined *Reduced Generalized Voronoi Graph (RGVG)* structure for exploration map. The RGVG is essentially the GVG without its weak meet points. By adopting the RGVG instead of the GVG, a number of meet points and edges decreases. With fewer edges and meet points, localization becomes more difficult. So, we developed an edge matching procedure for topological matching to avoid a useless edge matching and to maintain reliability.

2 Related Work

This work relates to two research fields, sensor based exploration and mobile robot localization. Both research fields are so vast that we only include the works that have influenced to our work.

2.1 Sensor Based Exploration

Sensor based exploration has a possibility to be used in various mobile robot applications, and many algorithms and techniques have been proposed.

A behavioral based exploration is a popular approach. The robot is armed with a simple set of behaviors (e.g., following a wall) to explore environment [4]. Pixel based exploration is also popular approach. A planar environment is discretized into pixels of some resolution (e.g., [5]). Strong experimental results verified the utility of these approaches. However, these heuristic approaches do not guarantee that a path can be found theoretically.

Oh the other hand, non-heuristic sensor based algorithms have been proposed in the plane (see [6] for a survey). They guarantee a completeness of finding a path theoretically, however many of them verified only in simulation and little experimental results in real world have been reported.

2.2 Mobile Robot Localization

Localization is a major topic in mobile robot research area, and many algorithms and techniques have been proposed. [7] is a complete survey of current localization techniques. Most of them are constantly trying to update the robot's (x, y) coordinates relative to a global frame by sensor information.

Our localization approach is different philosophy from them. It is trying to localize a robot location on a topological map without updating the robot's (x, y) location.

Dudeck[8] and Kuipers [9] have reported with the same philosophy with ours. In [8], a robot localizes on a graph by matching up nodes and the adjacency relation. However, the approach assumes the robot can label each node by depositing a marker at the nodes. Kuiper's work[9] is similar to our method. The robot essentially traces double equidistant line until a sensor threshold is met, then the robot follows the obstacle boundaries. Both algorithms verified in simulation, however it is not verified by real robot in real environment.

3 Basic Strategy of Exploration

The planar GVG is a set of points expressed by

$$G(x) = \big[(d_1 - d_2)\big](x) = 0 \tag{1}$$

where d_1 and d_2 are distances to two convex objects. Since G is a function of distance, a GVG edge can be computed from range sensors. Once the robot acquires the whole GVG structure, it has effectively all the relevant environmental information because every point in the environment is visible at least one GVG edge.

This section details the GVG incremental tracing technique:

1) Acessing GVG

First, the robot accesses GVG. This motion is realized by moving away from the closest object until it arrives at the point equidistant to two objects.

2) Tracing GVG

Once the robot gets on GVG, it traces GVG edge. The motion is realized by maintaining two equidistance between two closest objects. A tangent vector of local GVG edge that the robot should follow is determined as the vector perpendicular to the line that passes through two closest points of each object. Figure 1 shows a calculation example of tangent vector, which is expressed by outline arrow. A mathematical detail of the control law for tracing GVG is described in [2].

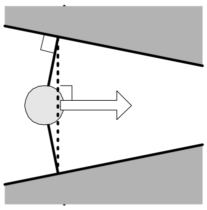


Figure 1: Tracing

3) Meet Point Detection

Several GVG edges meet at the point equidistant to three (or more) objects termed a *meet point*. When the robot arrives at the meet point, it stores the point location and branching edge angles, and continues to follow one of unexplored edges. If there is no unexplored edge associate with the meet point, the robot backtracks along an already explored edge to trace another unexplored edge. Once every meet point has no unexplored edges, exploration is done.

4) Boundary Point Detection

Some GVG edges terminate at the boundary of environment (e.g. corner), and the robot has only one choice to follow at the point. This point is called *boundary point*. When the robot arrives at the point, it simply backtracks the explored edge.

Simple example of above exploration procedure is illustrated in Figure 2.

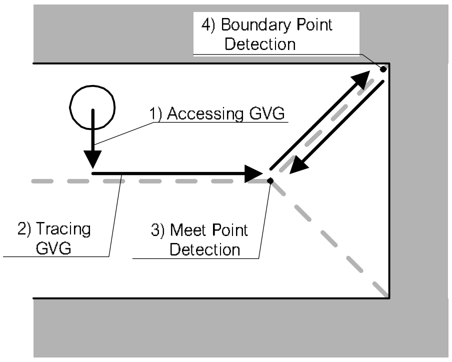


Figure 2: Basic Strategy of Exploration

4 Localization by Topology Matching

In our first attempt to implement above exploration algorithm on a real mobile robot, we used dead reckoning for localization. When the robot arrives at a meet point, it simply compares its current coordinates (calculated by dead reckoning system) with every registered coordinates of meet point. If pair of coordinates matches, the robot recognizes that it has arrived at an already visited meet point. This worked fine in a simulation and in small-sized environments (total run was less than 10 meters). However it failed many times in larger environments because of accumulated positioning error of dead reckoning.

To deal with this problem, we had proposed a localization approach using topological matching [3]. In this approach, when the robot arrives at new meet point, instead of looking at its current position, it looks at the current values of the sonar sensors. If this “sensor signature” matches the signature of previously encountered meet points, then current meet point is candidate for being an already explored meet point. The robot then explores a GVG edge until it reaches a neighboring meet point. If both current and neighbor meet points match pair-wise to meet points connected by an edge of the same length and intra-meet-point distance in the partially constructed GVG, then the candidate meet point (and its neighbor) is (are) more likely to be an already visited meet point. This procedure is repeated once more and if there is a match, the robot concludes that it has re-visited already encountered meet points. Note, that if multiple matches result, then the robot visits additional meet points until a unique graph match is obtained.

See Figure 3 for an example of this localization approach. First, the robot visits meet points 1, 2, and 3. Later on, in the course of exploration, the robot re-encounters these meet points, but initially cannot conclude their true identity, so the robot labels them meet points 8, 9, and 10. After graph matching, the robot identifies the true identity of the meet points and reduces localization error.

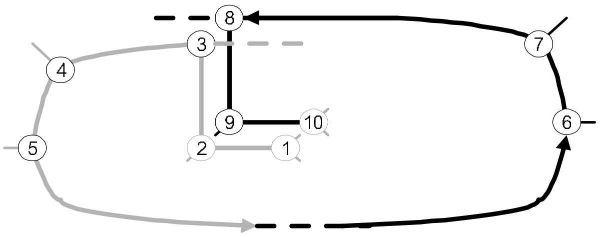


Figure 3: Localization by Topology Matching

We implemented this localization approach on a real mobile robot, and it successfully worked in some large environments, but the robot sometimes failed localization because of weak meet point problem.

4.1 Weak Meet Point

The above localization method essentially requires consistency of meet points’ relationship in GVG structure. If some meet points are unstable (sometimes the meet points appear and sometimes disappear), this localization approach can easy fail. Unfortunately, this is not a non-generic situation in real environments because sensing ability is often limited. We call such unstable meet points, *weak meet points*. Figure 4 shows a typical example of weak meet points. Both environments are the same and the left GVG structure is theoretically correct. However, sometimes the robot acts like a right figure, because it does not recognize the concavity in upper wall from its sensor readings.

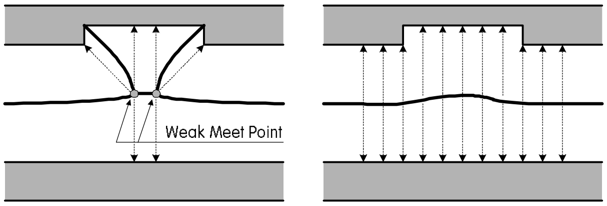


Figure 4: Weak Meet Point

4.2 Reduced Generalized Voronoi Graph

To deal with the weak meet point problem, we tried to remove every weak meet point from GVG, instead of trying to detect meet points more precisely.

Almost of all weak meet points are associated with edges that contact two objects, called *boundary edges*. In all cases, such edges do not have to be explored as they do not provide any topological information. Therefore we adopted a GVG structure without boundary edges, called the *Reduced Generalized Voronoi Graph (RGVG)*. Figure 5 shows a comparison example between conventional GVG and RGVG.

Note that RGVG is a unique structure that is a subset of the GVG. Also RGVG is connected because it is produced by eliminating boundary edges from GVG. Therefore, RGVG is qualified as exploration map of environment.

4.3 Edge Matching Technique

In the localization approach by topology matching shown in [3], we used only edge length information for edge matching. Actuality, this criterion is naive because different GVG edges could be matched accidentally. Furthermore, once we adopt RGVG structure for exploration, the localization procedure becomes very difficult because there are fewer meet points and hence fewer edges. We found that in our experiments that matching with meet point information and edge lengths is not enough. So, we started to consider other matching criterion. We assume that the robot stores

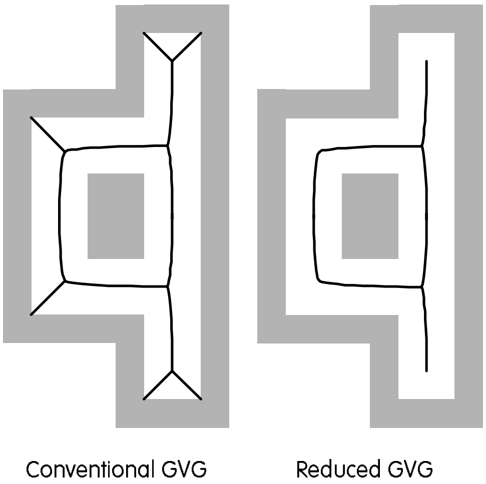


Figure 5: Reduced Generalized Voronoi Graph

(1) robot's coordinates from dead reckoning, and (2) distance and orientation of the two closest objects, in every sensor update. Then, in addition to edge length comparison, we use an edge shape comparison and edge area comparison for edge matching.

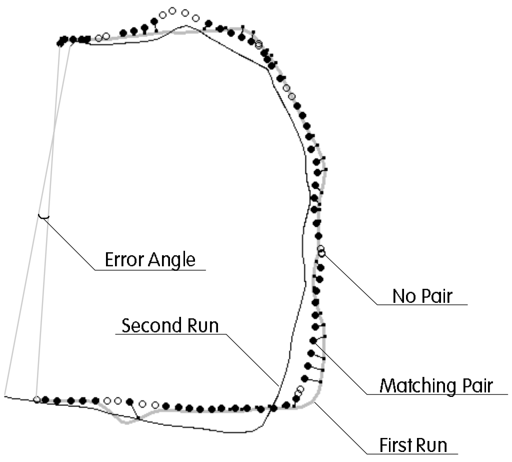
1) Edge Length Comparison

The simplest edge comparison scheme is to compare length. The edge length can be approximated by a sum of sampling points' distance, $df(E_1, E_2)$, where E_1 and E_2 are the set of sampling points in first and second edge.

If $df(E_1, E_2)$ is smaller than a conservative threshold (If both edge lengths are the same), matching procedure enters following step. Temporally, we set the threshold as 10 percent error of edge length.

2) Edge Shape Comparison

Before comparing edges, place both edges (their respective sequence of points) into a common coordinate system. So, one edge is translated and rotated to lie in the same coordinate system as the other edge. The translation vector is simply the displacement vector between the start points of both edges. Let p_{1s} be the start point of E_1 and p_{2s} be the start point of E_2. p_{1s} is also the origin of E_1. The displacement is simply $p_{1s} - p_{2s}$. Let p_{1e} be the end point of E_1 and p_{2e} be the end point of E_2. The orientation of the coordinate system is approximated by the angle between $p_{1e} - p_{1s}$ and $p_{2e} - p_{2s}$. This procedure is shown in Figure 6.



Explanation: The gray line represents the first edge and black thin line represents the second one in the same environment. The black dots and white small circles represent the rotated sampling point of second edge. (White circle means that it has no pair in first edge.) Parallel translation and rotation of the second edge compensate a comparison problem associated with the comparison.

Figure 6: Edge Shape Comparison

Now, E_2 is translated and rotated into the same coordinate system as E_1. We tried several techniques for comparing the edges.

First, we tried comparing points that are the same distance along the edge (integrate distance with respect to arc length). This did not work when some edges had a high frequency component, perhaps due to sensor noise. In other words, edges with a slight ripple in them would seem longer than other edges.

Then, we decided to compare points between edges by comparing the "closest pairs" of points. For each point in E_1, there is a closest point in E_2 and each of these pairs has a distance between them. If multiple points in E_1 map to the same point in E_2, we only consider the smallest distance of the pairs. Let n^* be the number of remaining pairs of points and let k be an index of remaining pairs. We then sum the remaining pairs to determine the distance between E_1 and E_2, i.e.,

$$cs(E_1, E_2) = \frac{\sum_{k=1}^{n} dist(p_{1k}, p_{2k})}{n^*} \qquad (2)$$

The denominator n^* normalizes the summation, so this result is independent of length. If $\frac{n^*}{n} < .8$, we do not use above procedure for matching. (Keep in your mind that n is the number of points on E_1).

3) Edge Area Comparison

The size of the free space in one edge is also independent from positioning error. Therefore, we defined a correlation function of edge area comparison $ca(E_1, E_2)$ using the remaining pairs of E_1 and E_2.

$$ca(E_1, E_2) = \frac{\sum_{k=1}^{n} abs\,(d_{1k} - d_{2k})}{n^*} \qquad (3)$$

where d_{1k} is a distance to the closest object at p_{1k}, and d_{2k} is a distance to the closest object at p_{2k}.

This is not exact a comparison of the size of free space area, however it must match when the robot traverses the same pair of edges.

Once $df(E_1, E_2)$, $cs(E_1, E_2)$, and $cf(E_1, E_2)$ are smaller than threshold, E_1 is matched with E_2. In our implementation, the threshold of $cs(E_1, E_2)$ is $5[inch/pair]$, and threshold of $cf(E_1, E_2)$ is $10[inch/length]$.

5 Experiment

We implemented the proposed localization approach on our Nomad 200 mobile robot with 16 sonar sensors in a laboratory environment using parameters mentioned in the previous section.

Figure 7 shows one experimental result of exploration in real environment (about $4 \times 6 meters$). The robot started exploration from meet point 1, traced a loop and came back to meet point 1 again. It labeled as meet point 4. Then it traced the same edge toward meet point 8 (because the robot did not conclude that it arrived at old meet point). In this second run, the robot found three weak meet points 5, 6 and 7. However, once it had enough sensor data, these weak meet points were deleted in this structure (just displayed). Then, the robot matched meet points and edges, and succeeded construction of RGVG in this environment.

6 Conclusion and Future Works

We succeeded to reduce a weak meet point problem by using RGVG, and this localization approach can be applied in large scale environment. Now the robot succeeds exploration task more than 90 percent in our laboratory size environment by this approach. However, we still have following problems.

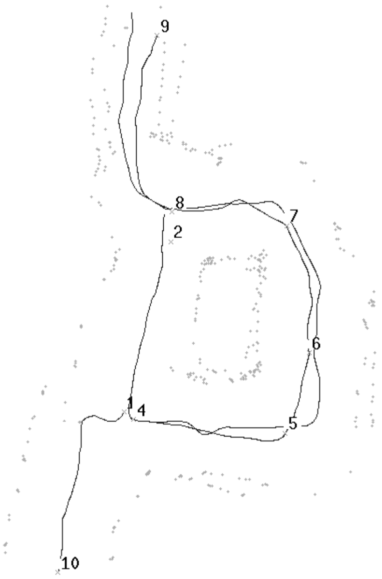


Figure 7: Experimental Results

1) Hyper Symmentries Problem

If an environment includes hyper symmetry, this approach fails because topology matching relies on the sensor signature of meet points and edge length, not (x, y) coordinates. For example, the robot fails to localize in Figure 8 because every meet point looks same and every edge length is same in GVG structure. Once we use RGVG in this example, no meet point appears in this example.

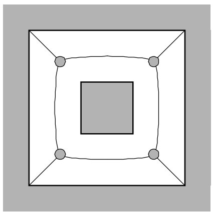


Figure 8: Hyper Symmetry

2) Sensor Range Problem

Once the robot recognizes only one object in its sensor range, the robot never traces a GVG line. Current sonar sensor's range is about 5 meters, so the robot can not explore in a large open space by current algorithm.

3) Future Works

Next step of our research is to solve above problems. Incidentally, this edge matching technique may be applied on a localization of multi-robots' exploration because the method is independent from positioning error. This is another application of localization technique based on edge matching.

References

[1] 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.

[2] H Choset, I Konuksven, and A Rizzi. Sensor Based Planning: A Control Law for Generating the Generalized Voronoi Graph. In *Proc. of IEEE Int. Conf. on Autonomous Robots*, Monterey, CA, 1997.

[3] Keiji Nagatani, Howie Choset, and Sebastian Thrun. Towards exact localization without explicit localization. In *Proc. of IEEE International Conf. on Robotics and Automation*, pages 342–348, 1998.

[4] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[5] J. Borenstein and J. Koren. Real-time obstacle avoidance for fast mobile robots in cluttered environments. In *IEEE Conference of Robotics and Automation*, pages 572–577, Cincinnati, Ohio, May 1990.

[6] 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.

[7] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques.* A. K. Peters, Ltd., Wellesley, MA, 1996.

[8] G. Dudeck, M. Jenkin, E. Milios, and D. Wilkes. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation*, 7:859–865, Dec. 1991.

[9] B. Kuipers and Y.T. Byan. A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations. *Journal of Robotics and Autonomous Systems*, 8:47–63, 1991.