Sensor Based Navigation for Car-like Mobile Robots Based on Generalized Voronoi Graph -Implementation and Experiment-

Keiji Nagatani   Yosuke Iwai   Yutaka Tanaka
The Graduate School of Natural Science and Technology, Okayama University

3-1-1 Tsushima-naka Okayama 700-8530, JAPAN
Phone : +81-86-251-8065
Fax : +81-86-251-8024
keiji@ieee.org

Abstract: Our research objective is to realize a sensor-based navigation by a car-like mobile robot in unknown environment. Generalized Voronoi Graph (GVG) is used for the navigation in this research. The GVG is a good tool for representing a topological map of environment, and it can be constructed by range sensor data incrementally. However, GVG is not suitable for a path of car-like mobile robots, because it is not always smooth. Therefore, the GVG should be modified for car-like robots.
In this paper, we represent implementation issues for path representation to enable a sensor based navigation for car-like robots. Experimental results support this method.

Keyword: Sensor based navigation, Generalized Voronoi Graph, Bezier curve

1 Introduction

Our research objective is to realize a sensor-based navigation by a car-like mobile robot in unknown environment. Generalized Voronoi Graph (GVG) is used for the navigation [1]. The GVG is a map embedded in robot's free space that captures the topologically salient features of a free space. Therefore, knowing the GVG is equivalent to knowing the free space, and constructing the map is kin to exploring the free space. A feature of GVG is that it can be constructed incrementally using data of range sensor. Therefore, it is a good tool for an exploration task of a mobile robot that has a range sensor. Unfortunately, a shape of GVG is not always smooth, and a car-like mobile robot can not follow it exactly because of minimum turning radius. To enable an exploration by a car-like robot based on GVG method, we proposed the following procedure to deform robot's path[2].

1. Constructing a local GVG by a range sensor
2. Determining a local goal on the local GVG
3. Generating path candidates expressed by Bezier curves that satisfies the following conditions.
   (a) The local goal and the robot's position are set as anchor points for a Bezier curve.
   (b) The maximum curvature of each curve should be smaller than the inverse number of minimum turning radius of the robot.
   (c) Each path must be non-collision with obstacles.
4. Choosing the best path using an evaluation function, and tracing it
5. Repeating 1-4

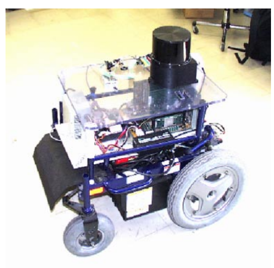

Fig. 1: Robot platform

To verify a validation of above method, we developed an autonomous car-like mobile robot with laser range finder (shown in Fig.1), and succeeded sensor based exploration in a corridor environment.

In this paper, we report implementation issues to realize sensor based navigation in real environment, particularly focusing on "Fast GVG acquisition" from laser range sensor's information. Usually a laser range sensor returns rich data, then a flat surface is recognized as a small bumpy surface because of sensing errors. Thus it generates very complicate GVG, and it takes very long time for calculation. To avoid this problem, we developed a fast GVG acquisition algorithm, which generates GVG from range data directly shown in section 4.

2 Related Works

Our research relates two research areas. Although both of these fields are vast, included works are the works that have influenced the authors' thinking.

2.1 Car-like robots' navigation

To generate a path for car-like robots, many heuristic's algorithms were proposed. To consider a non-heuristic algorithm, a simple one is to combine a set of line segments and arcs of circles [3]. It is guaranteed that a car-like robot can follow a planned path if the radiuses of the composed circles are bigger than minimum steering radius of the robot. The other approaches are "the shortest path for car-like robots in manifold" [4], and "nonholonomic distance" [5]. Both approaches aim to find the optimal path for car-like mobile robot with nonholonomic constraint.

Our approach is to generate candidates of Bezier curves as robot's trajectory based on GVG, but optimality is also considered[2].

2.2 Sensor based planning

A sensor based planning enables a robot to explore in unknown environment. Usually a conventional research for sensor based planning assumes simple and weak sensors (for example Bug algorithm [6]). However, in real environment, we believe that a robot should have more powerful sensors (like a laser range sensor) to enable sensor based navigation. In this case, the strategy may be different from above approach.

Choset et. al. proposed a sensor based planning method by an incremental construction of GVG [7]. It requires line of sight information only (that is obtainable by robot's range sensors) and the procedure has no restrictions of obstacles' shape. The algorithm has been successfully implemented on an actual mobile robot with a ring of sonar sensors [8].

Our approach basically follows as Chosets' work. However, our car-like robot must generate smooth path because GVG is not smooth.

3 Conventional Method based on GVG

This section reviews the exploring procedure for holonomic mobile robots that has been proposed in [7].

3.1 Definition of GVG

In a planar case, a target environment is categorized by objects and free spaces, and each concave object is divided into several imaginary convex objects. Then a planar GVG is defined as a set of equidistant points from the two or more closest objects, which is unique in static environment. An example of GVG structure is shown in Fig.2.

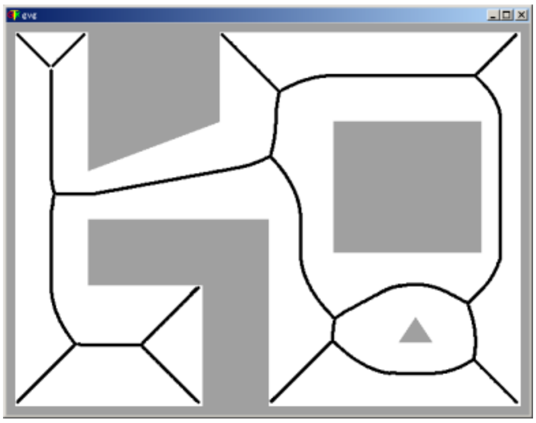

Fig. 2: An example of GVG structure

3.2 Exploration procedure

GVG can be constructed incrementally using mobile robots with range sensor by the following procedure.

1. **Tracing GVG**

   A motion for a mobile robot to follow GVG edge is realized by maintaining equidistance between two closest convex objects. A steering angle of the robot is perpendicular to a segment between the location of two closest objects' points. Mathematical details of the control law to trace GVG are described in [9].

2. **Detecting meet points**

   A meet point is, as its name suggests, a point where GVG edges meet. It is, at least, triple equidistance to the closest objects in planar case. When the robot detects a meet point, it stores the location of the meet point and branch directions of GVG. Then it traces an unexplored branch direction.

3. **Detecting boundary points**

   A boundary point is a point where a GVG edge contacts to objects. Therefore the robot must stop tracing GVG before it arrives at a boundary point, not to collide objects.

4. **Backtracking GVG**

   When the robot arrives at a point that is close to a boundary point, the robot can not continue to trace the GVG edge, then the robot must backtrack the edge to the last visited meet point. If there is no unexplored GVG edge associate with the meet point, the robot must backtrack the edge to find a meet point which associates with unexplored GVG edges. When there is no meet point that associates with unexplored GVG edges, the exploration is done.

3.3 Reduced GVG

Using above procedure, a robot may generate useless GVG edges for exploration in real environment. Weak meet points (it appears and disappears depending on sensing noise and threshold setting) are also troublesome. Fig.3-(A) shows an example of above problems. Two central meet points (weak meet points) may not be detected because of sensing noise.

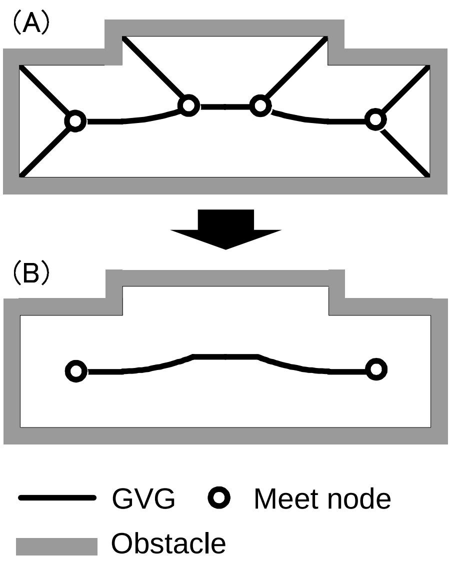

Fig. 3: Reduced GVG

Usually, these useless GVG edges are connected to boundary points, so Reduced GVG (RGVG) is defined by cutting such useless GVG edges from original GVG structure[10]. Fig.3-(B) is an example of RGVG structure reconstructed from the normal GVG structure shown in Fig.3-(A). This simple rule can avoid the weak meet point problem, and save costs for entire exploration in unknown environment.

Using above procedure, mobile robots can complete an exploration task by following GVG edges exactly. However, considering car-like robots, its path must be smooth (GVG is not). To solve this problem, we adopt following approach.

1. Detecting local environment information by a mounted laser range sensor
2. Generating local GVG
3. Planning smooth path using Bezier curve

To enable this procedure, we developed a fast GVG acquisition procedure from laser range data, shown in the next section.

4 Fast GVG Acquisition

In conventional approach shown in Section 3, a point on GVG is equal to a location of the robot, and whole GVG structure is constructed by robot's path. For a car-like robot's case, it is not suitable because of minimum turning radius of the robot. Therefore our approach is to construct a local GVG structure, and to deform it smooth for robot's path. One of the key to realize it is a fast acquisition of a local GVG. In this section, a GVG acquisition method from a laser range data directly is discussed using a simple experimental example.

4.1 Target area

A problem of a usual range sensor is that an accuracy of range data becomes worse according to a detection distance. Therefore, we exclude an outer range of fixed distance for generating local GVG. In our implementation, we set the threshold value as 5 meters. Fig.4 shows an example of target environment and valid range data by our laser range sensor. The sensor location is at the small circle, and the target area is defined as inside the large circle.

4.2 GVG point detection

A fine range sensor usually returns small bumpy data even if a surface of the object is flat. It generates many small GVG edges around the bumpy points. To avoid this problem, we use the following procedure to detect points on GVG edge (we call it as "GVG point").

1. Choosing an arbitrary point in free space, named "reference point".
2. Drawing a circle (named "search circle") with a reference point as the center of it, which intersects the closest point of the range data (named "object point").
3. Expanding the search circle to the direction of a gradient vector from the object point.

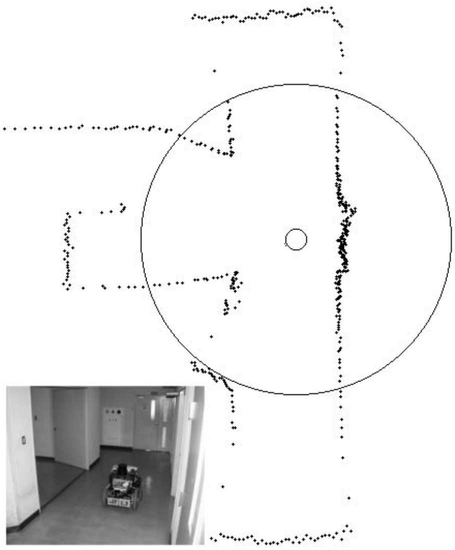

Fig. 4: Example of target area

4. If a number of "Same object group" inside the circle is one, returning to 3. "Same object group" is the group that a distance between two neighbor's range data is less than robot's size.
5. If a number of "Same object group" inside the circle is more than two, the center of the search circle is a GVG point.

Fig.5 shows a schematic of the procedure. The reference point is moving away from the object point until the other "Same object group" (dots of the bottom of the right) touches the search circle.

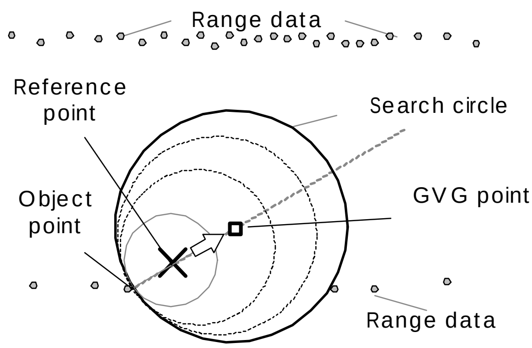

Fig. 5: Procedure of GVG point detection

4.3 GVG edge detection

If the above procedure is done at all points inside the target area, a local GVG acquisition can be completed. However redundant detection may be happened, and a calculation cost is too big. Also, a GVG edge is expressed by a set of non-sequential GVG points, which is difficult to use for a navigation of the robot. Therefore, we perform following procedure to detect GVG edges.

1. Finding range data gaps bigger than robot's size inside of the target area
2. Defining a center of the gap as a local GVG goal, and draw a segment between the current robot's location and the local GVG goal (named "reference segment")
3. Sifting a reference point from robot's location to a local GVG goal along the reference segment, and applying "GVG point detection" at each point, shown in Section 4.2.
4. Repeating 2 and 3 for every range data gaps

Using the above procedure, a GVG structure is expressed by a sequential series of GVG points, with low calculation cost.

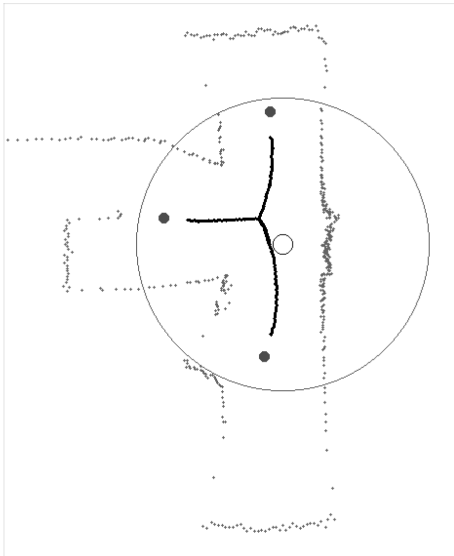

Fig. 6: GVG detection

Fig.6 shows an example of detected GVG edges, applied to Fig.4. In this figure, detected GVG edges are expressed as bold line, and each local goal point is expressed as small gray circle.

Another example in more complicate environment is shown in Fig.7. This example shows that the above method does not guarantee to generate full branches of GVG. That is because local goals depend on a range of target area. If a local goal is inside the range threshold, the goal and the GVG edge associated with it are disappeared. However, in our research experience, the proposed method is good enough to apply an exploration task for a car-like robot in usual environment (like a corridor environment).

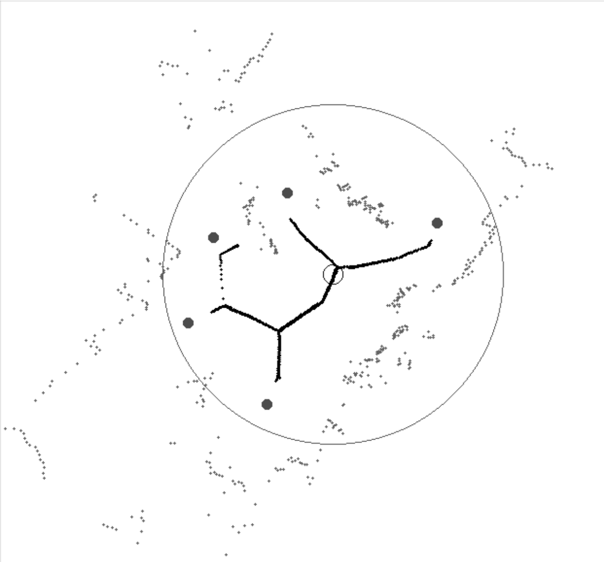

Fig. 7: GVG detection in a complicate environment

5 Generating Smooth Path

In Section 4, we proposed a local GVG construction method. A GVG edge is a skeleton of free space that the robot should follow. Therefore we set tips of local GVG as local goals. To connect robot's position and one of the local goals smoothly, we use the third order of Bezier curve.

A Bezier curve is defined by two anchor points $(x_0, y_0), (x_3, y_3)$ and two control points $(x_1, y_1), (x_2, y_2)$, shown in Fig.8. Then a point on the curve is defined as following equation, where u is a parameter from 0 to 1.

$$x = x_0(1-u)^3 + 3x_1(1-u)^2u + 3x_2(1-u)u^2 + x_3u^3 \quad (1)$$

$$y = y_0(1-u)^3 + 3y_1(1-u)^2u + 3y_2(1-u)u^2 + y_3u^3 \quad (2)$$

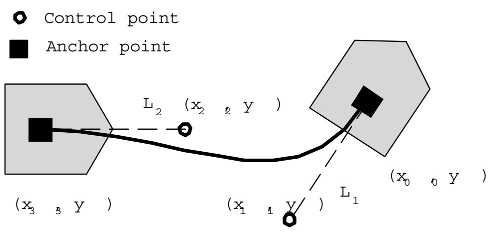

Fig. 8: A smooth path by Bezier curve

This is a feasible path for a car-like robot, because the tangent direction at (x_0, y_0) is equal to the robot, and it is differentiable at any point on the path. Once the tangent direction at (x_3, y_3) is set as the tangent direction of the GVG, unknown parameter is a distance between an anchor point and a control point (L_1 and L_2). The other words, we can generate candidates of smooth curve by changing L_1 and L_2. To choose the best path, we evaluate two conditions, (1) the path is smooth enough to trace by a car-like robot, and (2) the path is as far from obstacles as possible. A detail of the evaluation function is discussed in [2].

6 Experimental Results

6.1 Platform and sensor

Our research platform is a mobile robot based on a commercial electric wheelchair (shown in Fig.1). Steering angle of it is determined by a difference of rotational speed of two driving wheels automatically, and minimum turning radius of it is 80[cm]. Maximum velocity of it is 160[cm/sec], however we use 30[cm/sec] for safety. Each driving wheel has a rotary encoder to detect the wheel speed, and a position and orientation of the robot are estimated by wheels' rotational information (odometry system). In this research, the odometry system is only used for displaying data, not used for navigation. We use a robot controller as a standard PC (CPU : Celeron 450MHZ).

For range sensor, we use a laser range finder produced by Hamamatsu Photonics (C8074), the same sensor used in Section 4's example. It detects 500 range data in 360 degree in every 200[$msec$]. It is mounted at the center of the two driving wheels (a height of sensor detection face is 733[mm] from the ground).

6.2 Target environment

Target environment includes a corridor and laboratory rooms in Okayama University. A schematic of environment is shown in Fig.9. It is a standard indoor environment, and there is no object that can not be detected by a laser range finder. Surface of the ground is flat, so a wheeled mobile robot can move smoothly.

6.3 Experimental results

Fig.10 shows one of experimental results. Sensing data (gray dots) and robot's trajectory (bold line) is overlapped based on odometry information. The robot started at the lower right of this figure, and follow the path to upper right. Then it switched backward navigation (because it was dead-end), and continued following the GVG. The GVG acquisition and generating smooth path was done at gray circles on

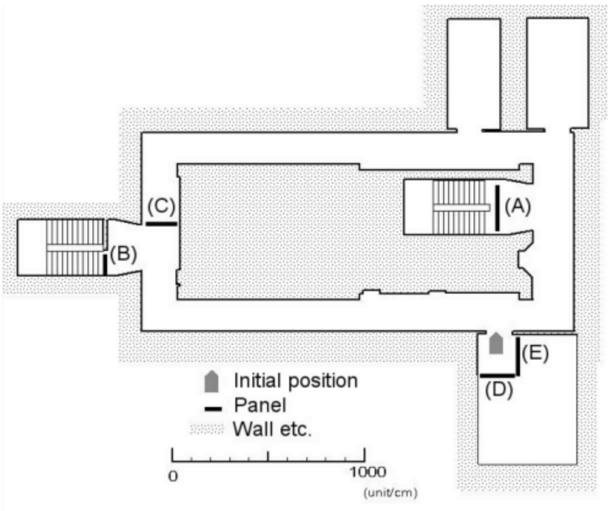

Fig. 9: Target environment

the trajectory. Finally, it arrived at the Goal location (the last position of exploration), which the same as the start position. Keep in your mind that the result figure is confused because it is plotted based on the odometry information, however it is no problem because it is completed topologically.

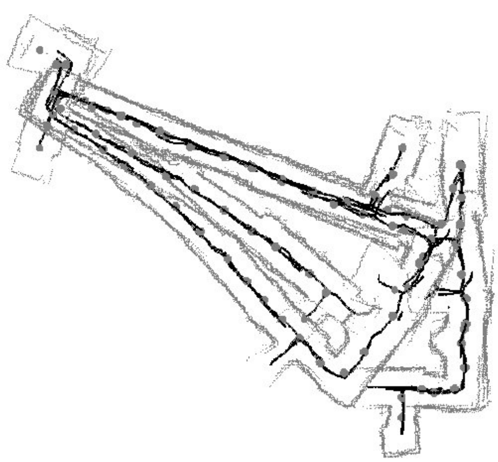

Fig. 10: An experimental result

7 Summary and Future Works

In this paper, we introduced a method to generate local GVG from laser range sensor, and an experimental result is introduced by the method.

In our experimental experiences, one of the problems is reflective objects. Laser range finder is difficult to detect reflective objects (such as glasses), and it prevents from constructing precise GVG. In this case, the other kinds of sensors are required for recognizing reflective objects. The other problem is very complicated environment. Our current algorithm is not applicable for such environment.

Our next step of this research is to solve above problems, and to perform a robust sensor based navigation in a complicated environment.

References

[1] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 2, pp. 125–137, 2001.

[2] K.Nagatani, Y.Iwai, and Y.Tanaka, "Sensor based navigation for car-like mobile robots using generalized voronoi graph," in *Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 1017–1022, 2001.

[3] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics 145(2)*, pp. 367–393, 1990.

[4] P. Moutarlier, B. Mirtich, and J. Canny, "Shortest paths for a car-like robot to manifolds in configuration space," *Int. Journal of Robotics Research, 15(1):36-60*, 1996.

[5] M. Vendittelli, J. Laumond, and C. Nissoux, "Obstacles distance for car-like robots," *IEEE Trans. on Robotics and Automation, 15 (4)*, 1999.

[6] V. Lumelsky and A. Stepanov, "Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape," in *Algorithmica, 2*, pp. 403–430, 1987.

[7] H. Choset, I. Konukseven, and J. Burdick, "Mobile robot navigation: Issues in implementation the generalized voronoi graph in the plane," in *Proc. of IEEE/MFI*, (Washington DC), 1996.

[8] K. Nagatani, H. Choset, and S. Thrun, "Towards exact localization without explicit localization," in *Proc. of IEEE International Conf. on Robotics and Automation*, pp. 342–348, 1998.

[9] 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.

[10] K. Nagatani and H. Choset, "Toward robust sensor based exploration by constructing reduced generalized voronoi graph," in *Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 1687–1692, 1999.