Communication

Yun-Hui Liu
Suguru Arimoto
Department of Mathematical Engineering and Information Physics
University of Tokyo
Bunkyo-ku, Tokyo, 113 Japan

Path Planning Using a Tangent Graph for Mobile Robots Among Polygonal and Curved Obstacles

Abstract

This article proposes a tangent graph for path planning of mobile robots among obstacles with a general boundary. The tangent graph is defined on the basis of the locally shortest path. It has the same data structure as the visibility graph, but its nodes represent common tangent points on obstacle boundaries, and its edges correspond to collision-free common tangents between the boundaries and convex boundary segments between the tangent points. The tangent graph requires $O(K^2)$ memory, where K denotes the total number of convex segments of the obstacle boundaries. The tangent graph includes all locally shortest paths and is capable of coping with path planning not only among polygonal obstacles but also among curved obstacles.

1. Introduction

The capability to autonomously plan a collision-free path among obstacles is one of the fundamental requirements for intelligent mobile robots. Path planning was popularly studied in the last decade, and many algorithms have been developed (for example, Lozano-Pérez and Wesley 1979; Brooks and Lozano-Pérez 1985; Khatib 1986; Noborio et al. 1988; and Liu and Arimoto 1990). The problem addressed in this article is described as follows: Suppose that in a two-dimensional environment a set O of obstacles with arbitrary boundary and a point mobile robot R having two degrees of freedom are given. The problem is how to find a path P for a robot from a start position S to a goal position G such that $R \cap O = \emptyset$ on any point $p \in P$.

Path planning of mobile robots among polygonal obstacles was first discussed by a visibility graph (V-graph) whose edges correspond to collision-free line segments between obstacle vertices (Lozano-Pérez and Wesley 1979). The visibility graph provides a compact and efficient scheme for path planning of point robots. However, the visibility graph requires $O(T^2)$ memory, where T is the total number of vertices of obstacles. It takes much calculation time to find the shortest path when T is large—in other words, when the robot is working in a work space with complicated obstacles. Moreover, the graph is limited to the case of polygonal obstacles, whereas most real-world obstacles have curved boundaries.

A generalized Voronoi diagram (GVD) gives a feasible approach for path planning in a general environment without any constraint on obstacle boundaries (O'Dunlaing et al. 1987; Takahashi and Schiling 1989). A generalized Voronoi diagram is the set of points that are equidistant from the boundaries of the obstacles and those of the work space. However, this method is subject to the drawback that the planned path is always as far from the obstacles as possible. Furthermore, its computation and representation of curved obstacles are difficult.

This article presents a new approach for path planning of mobile robots among obstacles with general boundaries by introducing a new graph called a *tangent graph* defined on a new concept: *the locally shortest path* (Liu and Arimoto 1990;1991). Like the local minimum in mathematics, a locally shortest path is defined as a path which is the shortest in a small local region. On the basis of the locally shortest paths, some special points called *tangent points* are selected on boundaries of the obstacles. In the tangent graph, a node represents a tangent point, and an edge corresponds to a collision-free common tangent between obstacles or a convex boundary segment between the tangent points. All locally shortest paths are included in the set of paths composed of edges from the node corresponding to start S to that corresponding to goal G. The tangent graph is capable of coping with path planning not only among polygonal obstacles but also

The International Journal of Robotics Research,
Vol. 11, No. 4, August 1992,

among curved obstacles. Furthermore, it has a smaller data structure than that of the visibility graph. In details, in a polygonal environment the tangent graph has $O(M^2 + N)$ edges, where M and N denote numbers of convex components and convex vertices of the obstacles, respectively. In an environment with curved obstacles the required memory is $O(K^2)$, where K denotes the total number of convex segments of the boundaries.

Laumond (1987) and Rohnert (1987) also discussed path planning by using tangents of obstacles[1] (also see Latombe 1991). Obstacles dealt with by Rohnert (1987) are convex polygons, and those by Laumond (1987) are generalized polygons that consist of circular arcs and line segments. This article differs in three ways from those by Laumond (1987) and Rohnert (1987). First, this article defines the locally shortest path and proves that the tangent graph includes all locally shortest paths. Next, boundaries of obstacles are not constrained to only arcs or line segments, and the proposed tangent graph is suitable for obstacles with any curved boundary. Third, we propose a new method for evaluating the data size of the tangent graph. Rohnert (1987) evaluated the data size by using convex components of the obstacles (see Theorem 2 in this article). However, we evaluate the data size by using convex boundary segments (spiral curves). In most cases, the number of convex segments of a complicated nonconvex obstacle is smaller than that of convex regions, so the proposed method gives more precise evaluation on the data size of the tangent graph.

2. Locally Shortest Path

The shortest path must be fundamental in motion of mobile robots, and how to find it is one of the most important problems in path planning. As in most optimization problems, detection of the shortest path requires considerable computation time. Instead of the shortest path, this article proposes another important kind of path called the *locally shortest path*, which is the shortest in a local sense. All locally shortest paths compose a subset of collision-free paths, and the subset can be readily found by a simple operation. Hence it is possible to design a more efficient algorithm for detecting the globally shortest path.

DEFINITION 1. Suppose that a curve C in a two-dimensional space D and a small positive real ϵ are given. The region

$$W = \{w : w \in D \text{ and } \mathrm{Dist}(w, c) < \epsilon, \exists \text{ point } c \in C\}$$

is called an *ϵ-neighboring region* of curve C, where $\mathrm{Dist}(w, c)$ is a function for computing the Euclidean distance between points w and c.

1. A reviewer pointed out that a similar idea already appeared in Laumond (1987) and in the Ph.D. thesis of Moravec at Stanford. Unfortunately those papers were not available to us. We developed the tangent graph at the end of 1989 independently and first presented the basic idea in Liu and Arimoto (1990).

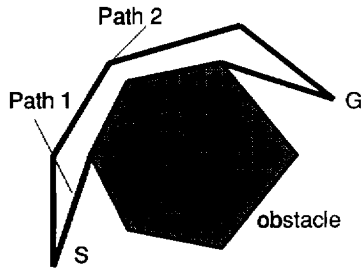

Fig. 1. Example of locally shortest path.

On the basis of the ϵ-neighboring region, the locally shortest path is defined as follows:

DEFINITION 2. A path is called locally shortest if there exists such a small ϵ that it is impossible to find a shorter collision-free path in its ϵ-neighboring region.

In Figure 1, path 1 is the locally shortest, but path 2 is not. Obviously, if no locally shortest path exists, no collision-free path exists in an environment and vice versa. The term *tangent* has a close relationship to locally shortest paths and will appear frequently in subsequent sections. Here we give it a distinct definition as follows.

DEFINITION 3. If a line contacts a point P on the boundary of an obstacle but does not intersect any internal point in a small neighboring obstacle region of point P, the line is said to be *tangent* to the obstacle on point P, and point P is called the *tangent point*.

DEFINITION 4. If a line is simultaneously tangent to two different obstacles, the line is called a *common tangent* of the obstacles. Furthermore, if small obstacle regions near the two tangent points lie in the same side of the line, the line is called an *external common tangent*; otherwise it is called an *internal common tangent*.

3. Path Planning Among Polygonal Obstacles

In general, most obstacles are polygonal regions or are approximated by polygonal regions. This section presents the tangent graph of polygonal obstacles and evaluates its data size.

3.1. The Tangent Graph

In a polygonal environment, any collision-free path of a point robot is composed of ordered line segments jointing start S to goal G via a sequence of obstacle vertices. The visibility graph $VG(V, L)$ is an undirected graph that registers all collision-free line segments connecting obstacle vertices, and it has $O(T^2)$ edges, where T is the total number of obstacle vertices. The efficiency of detection of the shortest path is strongly influenced by the number of edges. Consequently, when the obstacles have a complicated shape, finding the shortest path by the visibility graph costs much computation time.

The tangent graph $TG(V, L)$ has the same structure as the V-graph. However, its nodes and edges are defined on the basis of locally shortest paths. Node $v \in V$ represents an obstacle vertex on a locally shortest path, and edge $l \in L$ corresponds to a collision-free line segment, connecting two vertices, that is on a locally shortest path. The name *tangent graph* follows from the fact that its nodes and edges are tangent points and tangent segments of obstacles, respectively.

THEOREM 1. If the line segment connecting two vertices of obstacles is on a locally shortest path, the line extended from the segment must be tangent to the obstacles on the vertices.

Proof. Suppose that line segment CD is not a common tangent and a path $SCDG$ is planned (Fig. 2). If we can prove that the path is not the locally shortest, this theorem holds. Without loss of generality, assume that vertex C is not a tangent point. First, suppose that no obstacle vertex is on line segment CD. In this case, in any small ϵ-neighboring region of point C, it is always possible to find a point H on line segment SC such that line segment DH does not intersect any obstacle (Fig. 2A). Clearly path $SHDG$ is in ϵ-neighboring region of path $SCDG$ and shorter than path $SCDG$. Second, consider the case that finite vertices exist on line segment CD. Detect the closest vertex K to point C from such vertices (Fig. 2B). Similarly, in any small ϵ-neighboring region of point C, a point H such that line segment KH does not overlap any obstacle can always be found on line segment SC. Obviously path $SHKDG$ is shorter than path $SCDG$. □

This theorem states that a tangent graph for a given environment can be simply constructed by selecting collision-free common tangents between the obstacles. The process for checking whether a line is tangent to a polygon on a vertex is simple. If the line divides the internal angle of the polygon into two subangles at the vertex, it is not a tangent of the polygon and vice versa.

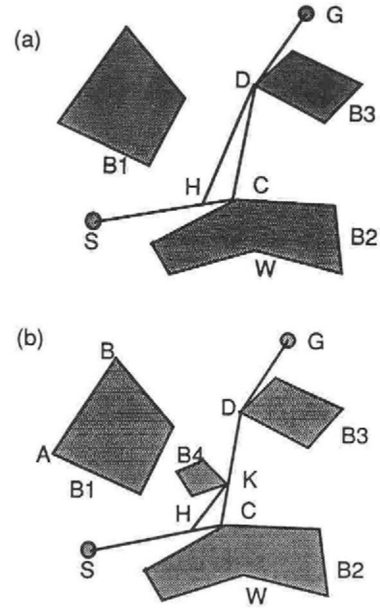

Fig. 2. Relation between locally shortest paths and lines between vertices.

3.2. Data Size of Tangent Graph

Obviously, from their definitions, the tangent graph has fewer edges and nodes than the visibility graph. To make a distinct difference between their numbers of nodes and edges, some theoretical evaluation is required.

PROPOSITION 1. For any two separated convex obstacles in a 2D space, there exist only four common tangents (Fig. 3).

Proof. From the definition of tangent, a tangent of a convex obstacle never intersects any internal point of the obstacle region and must contact the boundary of the obstacle. Furthermore, a line decomposes a 2D space into two half spaces. If a line is tangent to two convex objects simultaneously, the objects must lie in any one of the half spaces divided by the line. Because only four dispositions of two objects in two half spaces are possible, there exist

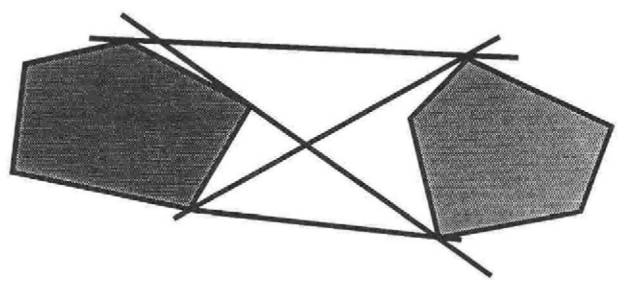

Fig. 3. There are only four common tangents between two separated convex obstacles.

only four common tangents for two separated convex objects. □

Definition 5. Given a vertex p, if the internal angle between two sides resulting in vertex p is smaller than π, vertex p is called a *convex vertex*; otherwise, it is called a *nonconvex vertex*.

From the definition of tangent, a nonconvex vertex is obviously never a tangent point. A nonconvex polygonal obstacle is a combination of a set of convex components. Clearly, a nonconvex polygon can be divided into infinite convex components. However, there exists a unique M such that the polygon cannot be divided into $M - 1$ convex components, but it can be decomposed into M convex components. Call M the *minimum number of convex components* of the polygon.

Theorem 2. M denotes the minimum number of convex components and N expresses the number of convex vertices of the obstacles. The data size of the tangent graph is $O(M^2 + N)$.

Proof. From Proposition 1, collision-free common tangents between M convex components are not more than $\frac{1}{2} \times 4M(M - 1)$. Moreover, a side between two convex vertices is a tangent of the obstacle, and the total number of such sides is not larger than N. Therefore the data size of the tangent graph for obstacles with M convex components is not larger than $2M(M-1)+N = O(M^2+N)$. □

It is simple to design an algorithm with two steps for computing the tangent graph for a set of polygonal obstacles. First, detect all common tangents between the obstacles. Next, remove all tangents intersecting any obstacle. Computation cost of the algorithm can be evaluated in the following way:

Theorem 3. Suppose that T denotes the total number of obstacle vertices and M and N represent numbers of convex components and convex vertices of the obstacles, respectively. The computation cost for the tangent graph is $O(N(N + T) + M^2T)$.

Proof. First, it is easy to check whether a line segment between two vertices is tangent to obstacles by using the sides resulting in the vertices, and hence the processing for line segments between N convex vertices costs $O(N^2)$ computation time. Next, because the intersection check between a tangent segment and obstacles with T vertices requires cost $O(T)$, it takes $O(T(M^2 + N))$ computation time to check intersections between $O(M^2 + N)$ tangents and the obstacles. Consequently, the algorithm runs in computation time $O(N(N + T) + M^2T)$. □

Computing a visibility graph of obstacles with T vertices takes $O(T^3)$ computation time, and hence it is possible to compute the tangent graph more efficiently than the visibility graph. Moreover, the visibility graph has $O(T^2)$ edges, but the tangent graph has $O(M^2 + N)$. To show such differences clearly, the visibility graph and the tangent graph are computed for the same environment (Fig. 4). The simulation results ascertain that the tangent graph has fewer edges than the visibility graph. Computing the visibility graph took 3.80 seconds, but computation time for the tangent graph was 0.58 seconds. The computation time includes graphic processing for drawing the graphs. The algorithm is programmed in C and is run in an Apollo DN 3500.

4. Path Planning Among Curved Obstacles

In the real world most obstacles usually have a complicated boundary like a continuous curve. Obviously, the visibility graph is not capable of coping with such a situation. However, the tangent graph can be simply extended to obstacles with an arbitrary boundary.

4.1. Tangent Graph of Curved Obstacles

On the boundary of an obstacle select a reference point o and define a coordinate $S(p)$ for representing position of a point p. The magnitude of $S(p)$ equals to the length of boundary segment between points p and o, where the length is counted around the obstacle clockwise. For a point p on the boundary and a small positive real γ, point

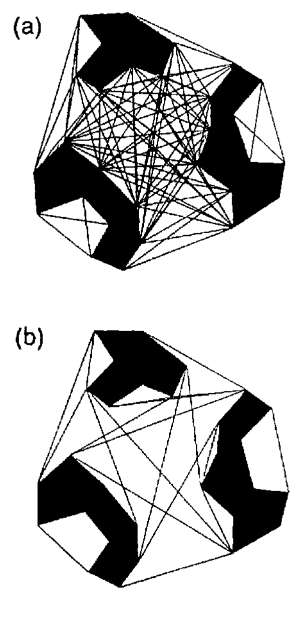

Fig. 4. A, The visibility graph. B, The tangent graph.

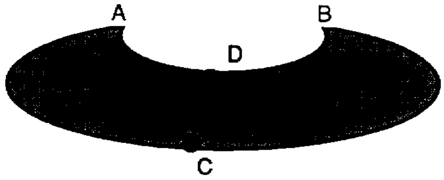

Fig. 5. Convex point C and nonconvex point D.

q on the boundary with $|S(q) - S(p)| < \gamma$ is called a γ-neighboring point of point p.

DEFINITION 6. Given a point p on the boundary of an obstacle. If there exists such a small γ that a line segment connecting any two γ-neighboring points of point p is always inside the obstacle, point p is *convex*. If the line is always outside the obstacle, the point p is *nonconvex*.

DEFINITION 7. If all points on a boundary segment C are convex, segment C is convex. If all points on the segment are nonconvex, it is nonconvex.

Figure 5 shows a convex point C and a nonconvex point D. In the figure, curved segment ACB is convex, but curved segment ADB is nonconvex. The boundary of an obstacle can be decomposed into a set of convex boundary segments connected by nonconvex boundary segments.

THEOREM 4. If a boundary segment is on a locally shortest path, it is convex.

Proof. It is easy to prove this theorem from definition of the locally shortest path, and therefore the details are omitted. □

THEOREM 5. If a line segment connecting two points on obstacle boundaries is on a locally shortest path, it is tangent to the boundaries.

Proof. This theorem can be proved in the same way as Theorem 1. □

The tangent graph is extended in the following ways: Its edge represents a collision-free common tangent of the obstacles or a convex boundary segment between two tangent points on a boundary, and its node corresponds to a tangent point on the boundary of an obstacle (Fig. 6). There are two differences between this tangent graph and that defined in the previous section. First, its nodes do not represent obstacle vertices, but tangent points of obstacles. Next, some of its edges correspond to curved boundary segments of obstacles.

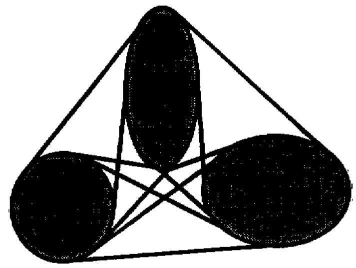

Fig. 6. The tangent graph for curved obstacles.

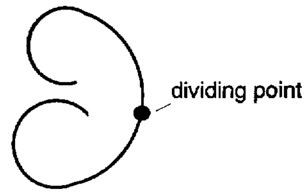

Fig. 7. Complexity of a convex boundary segment.

4.2. Evaluation of Data Size

The tangent graph is defined on convex boundary segments, and hence its data size is strongly influenced by complexity of convex boundary segments. A closed convex curve surrounds a convex region, and Proposition 1 states that only four common tangents are possible for two convex regions. Next, an open convex curve can always be divided into two curves with the property of a spiral at a point called the *dividing point* (Fig. 7). This article assumes that a spiral curve goes around only finite times.

For two spiral curves, three types of common tangents are possible. The extended line of tangent of type 1 never overlaps any spiral curve (Fig. 8A), and that of tangent of type 2 intersects one spiral curve (Fig. 8B), and that of type 3 tangent intersects both spiral curves (Fig. 9). It

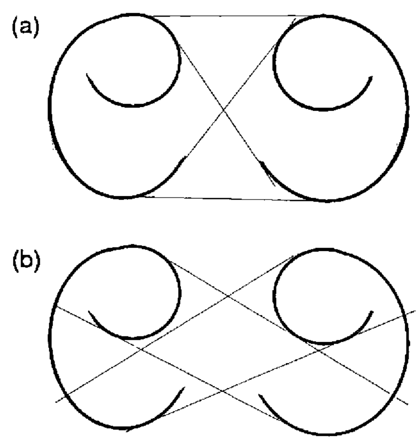

Fig. 8. Common tangents of type 1 (A) and type 2 (B).

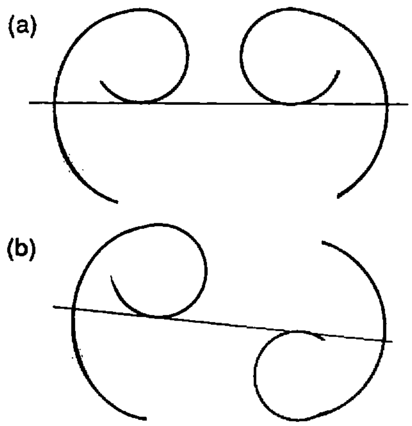

Fig. 9. Possible tangents of type 3 when two spiral curves can be separated by a line.

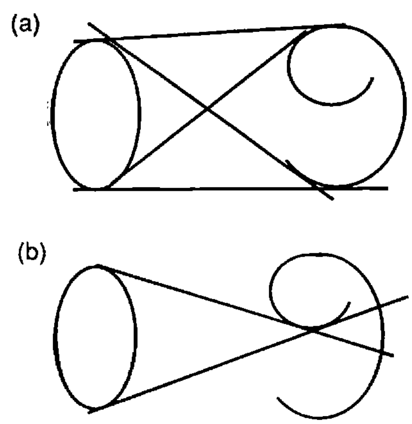

Fig. 11. Tangents between a spiral curve and a closed convex curve.

is easy to prove that there exist at most four tangents for each type in the same way as Proposition 1.

PROPOSITION 2. There exist at most nine collision-free common tangents for two spiral curves.

Proof. (1) First, consider the case that the two curves can be separated by a line. In this case, it is simple to prove that there exists at most an external common tangent (Fig. 9A) or an internal common tangent (Fig. 9B) of type 3.

(2) Next, consider the case that it is impossible to separate the two curves by a line. First, two internal common tangents of type 1 obviously are intersecting the curves. Next, suppose that four collision-free tangents of type 2 exist, and tangents 1,2 do not intersect curve 1 and tangents 3,4 do not intersect curve 2 (Fig. 10). Curves 1 and 2 must be inside regions 1 and 2 formed by the four tangents, respectively. Clearly, line AC separates the two curves. Consequently, at least one tangent of type 2 overlaps the curves. □

PROPOSITION 3. For a spiral curve and a closed convex curve, there exist at most six collision-free common tangents.

Proof. Two types of common tangents are possible for a spiral curve and a closed convex curve. First,

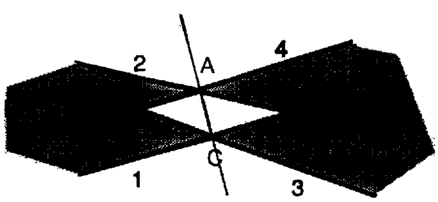

Fig. 10. Relation between regions of spiral curves and tangents of type 2.

the extended line of the tangent does not intersect the spiral (Fig. 11A). It is simple to prove that only four such tangents are possible in the same way as Proposition 1. Next, the extended line intersects the spiral curve (Fig. 11B). There exist two such tangents, because only two dispositions between the curves are possible. □

THEOREM 6. Suppose that K denotes the number of convex segments of obstacle boundaries. The tangent graph has $O(K^2)$ edges.

Proof. Suppose that the K convex segments are composed of H spirals and J closed convex curves. Clearly $H + J \leq 2K$. From Proposition 2, at most $9(H - 1)$ tangents can be drawn from a spiral i to other $H - 1$ spiral curves. Further, at most $6J$ common tangents are possible between spiral i and J closed convex curves from Proposition 4. According to the tangents, tangent points on spiral curve i are not more than $9(H - 1) + 6J$, and the spiral curve is divided into $9(H - 1) + 6J$ convex subsegments. The tangents about closed convex curves can be dealt with in the same way. The total number E of edges of the tangent graph satisfies

$$
\begin{aligned}
E \leq & \sum_{i}^{H} \frac{1}{2}[9(H-1)+6J] \\
& + \sum_{i}^{H}[9(H-1)+6J-1] \\
& + \frac{1}{2}\sum_{j}^{J}(6H+4J) + \sum_{j}^{J}(6H+4J) \\
= & \ 13.5H^2 + 18H \times J + 6J^2 - H \\
\leq & \ 13.5(H^2 + 2H \times J + J^2) \\
= & \ 13.5(H+J)^2 \\
= & \ O((2K)^2) = O(K^2).
\end{aligned}
$$

□

5. Finding a Path

The tangent graph $TG(V, L)$ consists of a set V of nodes and an edge set L connecting the nodes. An edge corresponds to a collision-free common tangent or a convex boundary segment. The path planning is the detection of a set of ordered edges connecting the node corresponding to start S to that corresponding to goal G. A direct method for this purpose is to search the tangent graph from node S one by one until the goal G is detected.

One method to find the globally shortest path is to enumerate all possible locally shortest paths and choose the one whose length is the smallest. However, much computation cost is required for this processing when the tangent graph has many edges. The A^* algorithm proposed by Hart et al. (1968) is an efficient heuristic method for minimizing the length of paths by evaluating the distances between nodes of the tangent graph. The details of the algorithm will not be dealt with in this article, but the reader can refer to the papers by Hart et al. (1986); and Lozano-Pérez and Wesley (1979).

6. Conclusion

This article proposed a tangent graph for path planning of mobile robots among general obstacles in a two-dimensional work space on the basis of a new concept: *locally shortest paths*. Unlike the visibility graph, the tangent graph, whose edge corresponds to a collision-free common tangent of the obstacles or represents a convex boundary segment of the obstacles, can be successfully applied to path planning among obstacles with arbitrary boundary. This permits more accurate path planning among curved obstacles without errors caused by polygonal approximation. Furthermore, its data size is mainly determined by the number of convex segments of obstacle boundaries, and hence it requires less memory than the visibility graph. Therefore the tangent graph provides a more powerful tool for path planning of mobile robots than the visibility graph, and its various applications are expected.

Acknowledgment

The authors thank their colleagues in the Arimoto laboratory and the reviewers for their helpful comments.

References

Brooks, R. A., and Lozano-Pérez, T. 1985. A subdivision algorithm in configuration space for findpath with rotation. *IEEE Trans. Sys. Man, Cybernet.* SMC-15(2):224–233.

Hart, P., Nilsson, N. J., and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Sys. Sci. Cybernet.* SSC-4:100–107.

Khatib, O. 1986. Real time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* 7(1):90–98.

Latombe, J. C. 1991. *Robot motion planning.* Boston: Kluwer Academic Publishers.

Laumond, J. P. 1987. Obstacle growing in a non-polygonal world. *Information Process. Lett.* 25:41–50.

Liu, Y. H., and Arimoto, S. 1990. A flexible algorithm for planning local shortest path of mobile robots based on reachability graph. *Proc. IEEE International Workshop on Intelligent Robots and Systems*, pp. 749–756.

Liu, Y. H., and Arimoto, S. 1991. Proposal of tangent graph and extended tangent graph for path planning of mobile robots. *Proc. IEEE International Conference on Robotics and Automation*, pp. 312–317.

Lozano-Pérez, T., and Wesley, M. A. 1979. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun.* ACM 22:560–570.

Noborio, H., Naniwa, T., and Arimoto, S. 1988. A fast path planning algorithm by synchronizing modification and search or its path-graph. *Proc. IEEE International Workshop on Artificial Intelligence for Industrial Application*, pp. 351–357.

O'Dunlaing, C., Sharir, M., and Yap, C. K. 1987. Retraction: a new approach to motion planning. *Planning, Geometry and Complexity of Robot Motion*, pp. 193–213. Norwood, NJ: Ablex.

Rohnert, H. 1987. Shortest paths in the plane with convex polygonal obstacles. *Information Process. Lett.* 23:71–76.

Takahashi, O., and Schiling, R. J. 1989. Motion planning in a plane using generalized Voronoi diagrams. *IEEE Trans. Robot. Automat.* 6(2):143–150.