Two-Disk Motion Planning Strategy

by

Yangsheng Xu, Raju Mattikalli, Pradeep Khosla

EDRC 24-46-91

TWO-DISK MOTION PLANNING STRATEGY [1]

Yangtheng Xu, Raju Mattikaffi, Pradeep Khosla

**Engineering Design Research Center
and The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213**

ABSTRACT

This paper addresses the problem of planning the motion of a polygonal object through a set of planar obstacles. We propose a *two-disk motion planning strategy* to navigate the object within the free space between the obstacles from an initial location to a final location. This method makes use of the Medial Axis Transform (MAT) of the free space which can be generated efficiently using the method developed in [10]. We determine two minimal overlapping disks that fully enclose the moving object, and then constrain the centers of the two disks to move continuously along a path on the medial axis. In this paper we direct our efforts to the problem of finding the enclosing disks for a moving object which is considered as a polygon. We formulate the problem as an optimization problem in terms of the geometry of the polygon to determine a local optimum for a particular edge pair. An algorithm to identify a global optimum from the local optimization result is proposed. Simulations are presented for a variety of polygons. The method is being used for disassembly motion planning of a subassembly within its parent subassembly.

1   Introduction

Planning the motion of an object (or objects) among obstacles has been a much researched topic in robotics. The *find-path* problem is one of finding a path for navigating an object among obstacles from its initial position (and orientation) to a final one such that there is no intersection with the obstacles. Our interest in motion planning is motivated by our effort towards assembly planning using detailed geometric models. It is desired to plan the disassembly motion for a part/subassembly from its initial position and orientation inside an assembly of parts to a position outside the assembly.

A variety of methods to solve the find-path problem have been proposed [3, 2], differing greatly in the representation schemes that are employed, computational complexity with respect to both space and time, as well as their underlying assumptions and limitations. In the configuration space approach [6, 7] the original problem of planning motion through a space of obstacles is transformed into an equivalent, but simpler problem of planning the motion of a point through a space of enlarged configuration space obstacles. In the freeways approach [1], the free space is explicitly represented by overlapping generalized cones called freeways, while translation is performed along freeways with rotations at the intersections of freeways. However, because of their computational complexity, these methods are fairly difficult to implement in realistic problems, with translation and rotation combined in a continuous matter.

Rather than attempting to use a single algorithm to solve the find-path problem in its full generality, we believe that, in planning disassembly motions, a combination of methods should be used; each generating a portion of the motion plan, depending on the kind of obstacles and the geometry of the moving part. Typically in a disassembly operation, the initial motions occur with moving components maintaining contact with their obstacles (the stationary components). However, subsequent motion in the same operation does not have obstacles close enough to maintain contact with the moving components. Under conditions of motion with contact, we use a motion planning method that exploits the contact conditions. In this paper, we present a motion planning strategy which is useful when the moving subassembly is not in close proximity with the stationary subassembly. We use a compact representation of free space and is ideal in situations of uncluttered obstacles.

We shall describe our algorithms for two dimensional objects, which are restricted to be polygonal. The statement of the problem is as follows:
Given:

1. a polygon V_o (which represents the obstacle space),

2. a polygon V_m (which represents the moving object) with an initial orientation and position ($\pounds_;$), such that P_m is within P_o, and

3. a final position for V_m ($\pounds/$)

find a motion plan for V_m from $\pounds_;$ to $\pounds/$, constraining V_m to lie within V_o at all times.

The free space within V_m is represented by its *Medial Axis Transformation* (MAT) [4, 5, 9, 8], which is used to plan the motion for V_m. A MAT of a certain region consists of two parts: (a) a medial axis (MA), which is the locus of points which are equidistant from two or more elements of the boundaries of the region, and (b) a radius function, which is the distance of each point on the MA to its nearest boundary element. This transformation is ideally suited for motion planning as the medial axis can be used to generate a path for motion and the radius function to ensure that collisions are avoided. A method to generate the medial axis for any arbitrary polygon is developed

in [10]. The connectivity between segments of the MA is represented in the form of a graph which is augmented with geometric information such as the length of the segments and minimum value of the radius function along a segment. An optimal path is then found on the graph giving us a path for moving V_m from £* to £ /.

Given a path on the MA within V_m, we present a method, referred to as the two-disk method, to generate a collision free motion plan for V_m from its initial to its desired position. The method determines two minimal overlapping disks that completely cover V_m. The centers of these disks are then constrained to move along the given path on the MA. A collision free path is ensured by requiring that the minimum value of the radius function along the path be no less than the smaller of the two radii of the disks.

The problem of determining two minimal disks is solved by calculating a straight line cut through V_m such that the two circumscribing disks around each of the generated sub-polygons represent a minimal set. We first prove that if the cut is optimal, the resultant two minimal disks have equal radii. We formulate the problem as an optimization problem for determining a local optimum for a certain edge pair of the polygon. A strategy is proposed to identify the global optimum from the results of each local optimization.

The method is efficient for a convex polygon, while for a concave polygon an additional computation is necessary. We present two intuitive ways to potentially reduce the computation. As· an alternative, we propose to approximate a concave polygon by a hypothetical convex polygon, i.e., the minimal convex polygon covering the given concave polygon. We have proved that the maximal radii of the disks of the hyperthetical convex polygon is 1.25 times of the redii of the disks that cover the original concave polygon. This implies that the approximation deos not result in too conservative solution.

The rest of paper is organized as follows. The two-disk motion strategy is introduced, and some associated theorems are stated and proved in the Section 2. In the Section 3, an algorithm is proposed for determinating two minimal disks for an arbitrary convex polygon, which includes three sub-problems, local optimization formulation, global optimization search, and an extension to the concave polygon case.

2 Two-disk Motion Planning Method

This section discusses the problem of planning a motion of an object V_o , based on the medial axis within the obstacle polygon V_o. We may recall the definition of the medial axis, that is, the locus of points which are equidistant from two or more boundaries of the polygon V_p. It can also can be viewed as the locus of centers of circles which make contact with the boundary at at least two points.

From the definition of the medial axis, we may consider the planning problem as the following way. If we determine a *single* enclosing circle for polygon V_{mi} i.e., a disk that fully covers V_m, and constrain the center of the circle to follow the global path, we are guaranteed collision free motion, provided the radius of the disk is no greater than the smallest value of the radius function along the path. For objects whose aspect ratio, that is a ratio of the largest value of the distance between two convex corners to the smallest value, is close to 1, fitting a single disk to circumscribe the object is an acceptable strategy. However, for objects whose aspect ratio is greater than 1, the single disk is far too conservative an approximation of the object. Naturally, if we attempt to circumscribe multiple overlapping disks over the polygon, the radii of these disks will be smaller than the single circumscribing disk. Surely, if an infinite number of disks are permitted, they would cover the object *snugly,* thus representing the exact size of the object.
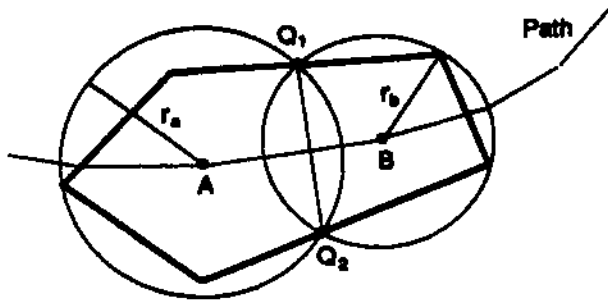
Figure 1   Motion planning on the global path by determining two minimal
overlapping disks to cover the given movable object and
constraining two centers of disks to follow the path

However, if multiple overlapping disks are used, a collision free path cannot be guaranteed ·
because *we* cannot constrain every disk to lie on the MA at all times. Thus, although multiple
disks represent a better approximation of the moving object, they cannot be used to generate valid
motion plans using our method of constraining disk centers to MA points.

It is not difficult to see that no more than *two* points fixed on a rigid object can lie on a given
2-D arbitrary path simultaneously. Considering the two centers of the enclosing disks as two points
on the movable object, one may see that maximally only two enclosing disks can be used for a
given arbitrary polygon, such that the centers of the disks kiss the given path. This is the rationale
of the so called **two-disk strategy** to navigate the given polygon V_m out of the V_o.

The two disk method plans the motion oiV_m with in V_o by enclosing V_m within two overlapping
disks and moving the disks along the path such that their centers are constrained to lie on the path
at all times. This method ensures collision free motion (so long as the smallest radius along the
path is no less than the two radii) by virtue of the fact that V_m is <u>completely</u> enclosed by the two
disks, and each disk individually is guaranteed to lie completely within the boundary of V_o. (see
Figure 1).

Therefore, we may state our problem as follows:

*Given an arbitrary polygon (S_m) find two minimal overlapping disks (D_1, D_2) such
that the polygon is fully covered by these two disks, i.e.,*

find

$$\min \left[\max(r_1, r_2) \right] \tag{1}$$

s.t.,

$$S_m \in Di \cup D_2 \tag{2}$$

where $r\backslash$ and r_2 are radii of the disks $D\backslash$ and IV
The solution can be obtained by solving the following problems.

*Given a 2-D polygon, find a **line** which splits the polygon into two polygons such that
the resultant subpolygons have minimum diameters for their circumscribing disks.*

4

**Figure 2   Polygon Sm is covered
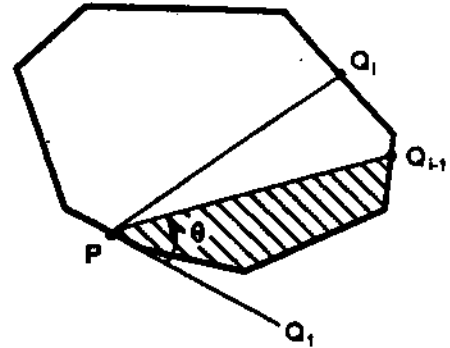by two minimal disks**

Figure 3   Optimal cut of a polygon

Then, for each resulting subpolygon, solve the following problem.

Given a 2-D polygon, find its smallest circumscribing disk.

As will be shown later, we actually solve the above two problems simultaneously. Before we develop an algorithm, let us investigate the first problem referred to as the *cutting problem* (see Figure 2), in more detail. In what follows, we prove that if the cut is optimal, then *these two minimum disks have equal diameters.*

On the boundary of the given polygon (shown in Figure 3), an arbitrary point P (referred to as the base point) is chosen. The polygon is cut into two along line PQi, where Qi is an other arbitrary point on the boundary and referred to as the *opposite point.* This cut results in two smaller polygons, the left polygon lying on the left side of the line, and the right polygon lying on the right side. The corresponding circumscribing radii are denoted by Ri and R_r. As the point Qi is varied in an anticlockwise direction starting from $Q\backslash$, polygons with different Ri and R_r are generated. When $0 = 0$, i.e., the cut is along the tangent to the boundary,

$$Ri \quad = \quad RQ \qquad \qquad (3)$$
$$R_r \quad = \quad 0 \qquad \qquad (4)$$

where RQ is the radius of the circumscribing disk covering the original polygon. For an arbitrary cut PQ_i,

$$0 \quad \leq \quad Ri \quad \leq \quad RQ \qquad \qquad (5)$$
$$0 \quad \leq \quad Rr \quad \leq \quad RQ \qquad \qquad (6)$$

For any two cuts PQi and $PQ\%-u$ *h* area of the right polygon with the smaller *9* is the subset of the area of the one with the larger *0*. As illustrated in Figure 3, the right polygon generating from the cut PQi includes the one generated from the cut $PQi-\backslash$. Therefore, the function R_r is monotonic increasing from 0 to RQ. Similarly, the radius function of the left polygon Ri is

5

monotonic decreeing from Ro to 0, with the increase of the cut angle 8. Our problem is to find a cut such that the larger of $J\!f\!y$ and Rr is a minimum, i.e.,

$$\min_{\theta}[\max(R_{ly}, Rr)] \tag{7}$$

This is equivalent to the problem of minimizing the difference between Ri and Rr.

$$\min_{t}[|A - JRr|] \tag{8}$$

The function $|Rt - Rr|$ first decreases from Ro, and then increases to Ro. Due to the monotone behavior of $R|$ and R_T, we expect that there exists a point such that the function $|Rt{\sim}R_r|$ vanishes, i.e., $R|$ is equal to R_r. This radius is denoted by $Ropt$ and the corresponding cut is an optimal solution. Therefore, we have

> **Theorem** 1: *The two minimal disks that cover a given arbitrary polygon are of equal diameter.*

Theorem 1 is applicable not only for a convex polygon, but also for a polygon that contain! concave corners, since the monotonic behavior of Ri and Rr which is due to the monotonic behavior of the areas of the smaller polygons, is still true. The only difference is that in the concave case, the derivative of the function $|Rt — R_r|$ is not continuous. Theorem 1 allows us to simplify the problem of finding two minimal disks to the problem of finding two identical disks.

It is noted that the optimal cut and corresponding radius $Ropt$ is for the point P. When the point is moved along the boundary, the corresponding optimal radius is different. Suppose an optimal cut is obtained based on point $P^{\bullet}$ (see Figure 4), i.e., $P^{\bullet}Q\%t^*$ Let the base point $P^{\bullet}$ be perturbed to a point P^{l} in the anti-clockwise direction. We will show that the optimal point $Q^{\wedge}{}_{t}$ corresponding to P^{l} always lies in an anticlockwise direction away from $Qopt^*$

When the base point $P^{\bullet}$ is moved away to the point P^{l} in the anti-clockwise direction, the opposite point could move in an anti-clockwise direction (i.e., to $Q\backslash pt)^{\wedge}$ or in a clockwise direction (to $Qlpt)$, or remain at the same point $Qopt$- Since the cut $P^{\bullet}Qopt$ is the best cut for the point $P^{\bullet}$, it is also the best cut for the point $Qopt$. This means the cut $P^{l}Qopt$ is not the best cut. Since Ri for the cut $P^{l}Ql_{pi}$ is smaller than $R|$ for the cut $P^{l}Qopu$ and R_r for the cut $P^{l}Ql_{pi}$ is larger than Rr for the cut $P^{l}Qopu$ the function $|Ri - Rr|$ for the cut $P^{l}Qopt$ is obvious smaller than that for $P^{x}Ql_{pt}$. Thus, the cut $P^{x}Ql^{\wedge}$ is not better than $P^{l}Qopu$ and thus is not the best cut. Therefore, an optimal cut at the base point P^{l} only could lie on in the anti-clockwise direction to $Qopt$, i.e., at $P^{l}Qlpv$

> **Theorem 2:** *// the base point is moved in an anticlockwise (or clockwise) direction, the opposite point also moves in the same direction.*

Theorem 2 allows us to search for the optimal cut in a single direction, thus reducing computational time.

3   Determination of Two Circumscribing Disks

This section addresses the problem of determining two minimal circumscribing disks for an arbitrary polygon. As mentioned in the previous section, the *optimal cut* is a function of the base point. When the base point moves along the boundary, the radii and location of the resultant optimal
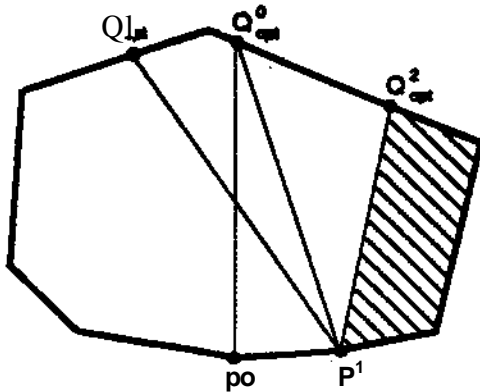
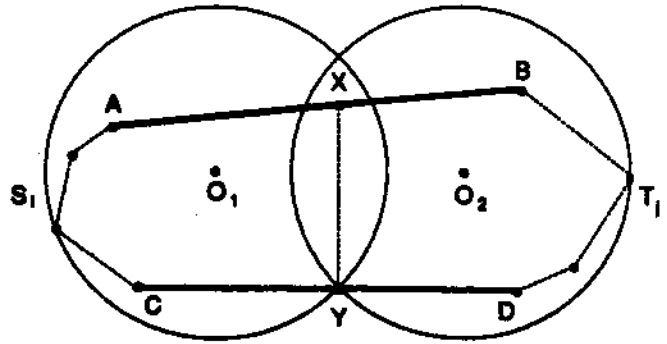**Figure 4   Illustration of Theorem 2**   **Figure 5   Optimal cut**

disks varies. Therefore, an optimal cut determined at a certain point or over a certain edge is a local optimum. A straightforward way to obtain the global optimum (i.e., the minimum over all edge pair) is as follows.

First, for any two edges of a polygon *(E% and Ej₉i ^ j)* (referred to as *edge pair),* optimization is performed to obtain an optimal cut over the edge ·pair, such that the resultant radius of the circumscribing disk is minimal. This process is then repeated for all possible edge pairs to obtain a global optimum. For a polygon with n edges, the number of possible edge pairs is $n(n-1)/2$. Thus local optimization must be performed $n(n-1)/2$ times in order to determine the global optimum by enumerating all possible edge pairs.

To reduce computational effort associated with generating optimal cuts for all possible edge pairs, we propose the following method. Select the edge pair where a global optimum is most likely occur, and identify the global optimum based on the geometry conditions. In what follows, the section 3.1 describes the formulation of the local optimization problem. In the section 3.2, we present a method for obtaining the global optimal solution. Extension of the method to concave polygons is discussed in the section 3.3.

3.1   **Formulation of the Local** Optimization **Problem**

Consider the convex polygon as shown in Figure 5 (an extension to concave polygons is presented in section 3.3). A cutting line intersects the boundary of the polygon at no more than two points. These two intersected points lie within or on both the disks. This means we will always find at least two points on the boundary such that they are common to both circumscribing disks. Consider the base point X on edge AB in Figure 5. Let XY be the optimal cut where the opposite point Y lies on the edge CD. Since X and Y are common to both disks, the distance from these two points to centers of both disks O_1 and O_2 should be less than the radius of the disk.

$$d(X, O_i) < r \tag{9}$$

$$d(X, O_2) < r \tag{1Q}$$

7

$$d(Y, O_x) \leq r \qquad (11)$$

$$d(Y_t O_7) \leq r \qquad (12)$$

where $d(P,Q)$ is the distance function from the point P to Q, and r is the radius of the disk to be optimized. Since the polygon is convex, the vertices of the polygon can be grouped into two groups; those that are contained within the left disk $(0i, r)$, and those in the right disk $(0_2, r)$, (see Figure 5). The vertex for the left disk is denoted by 5_i $(1 \leq i \leq m)$ where m is the number of vertices within the left disk, and the vertex for the right disk is denoted by Tj $(I \leq j \leq n)$, where n is the number of vertices within the right disk. The condition that each vertex lies within its corresponding disk is expressed as follows.

$$d(S_{it} Oi) \leq r \qquad (1 \leq t \leq m) \qquad (13)$$

$$d(T_{jt} O_3) < r \qquad (1 \leq j < n) \qquad (14)$$

Also, the points X and Y lie on edges whose equations are given by

$$A(*, y) = 0 \qquad (15)$$

$$l_2(x, y) = 0 \qquad (16)-$$

respectively. The radius of the disk r is the variable that is to be minimized. Thus we formulate a nonlinear optimization problem as follows.
Find

$$\min(r) \qquad (17)$$

s.t.,

$$d\{X, O_x) < r \qquad (18)$$

$$d(X, O_2) < r \qquad (19)$$

$$d(Y, Oi) < r \qquad (20)$$

$$d(Y, O_2) < r \qquad (21)$$

$$d(Su Oi) < r \qquad (1 < t < m) \qquad (22)$$

$$d(Tj, O_2) < r \qquad (1 < j < n) \qquad (23)$$

$$l_i(*, y) = 0 \qquad (24)$$

$$l_2(x, y) = 0 \qquad (25)$$

where $Oi(x,y)$, $Oa(x,y)$, $X(x,y)$, $Y(x,y)$, and r are 9 unknown variables. We have $m + n + 4$ nonlinear inequalities and 2 linear equation constraints for the solution of 9 unknown parameters. Since, the minimal number of vertices for a polygon is 3, i.e., $m + n \geq 3$, thus for this case we have 7 inequality and 2 equality constraints, to obtain 9 optimization variables. Thus the problem normally is solvable.

Let us take a rectangle in Figure 6 as an example. We first consider AB and DC as a pair of edges on which an optimal cutting occurs. Definition of the unknown variable set is as follows:

$$P = [r, O_1(x_1, yi), O2(x_2, y_2), X(x_3, y_3), Y(x_4, y_4)]^T \qquad (26)$$

The variable to be minimized is r, radius of the disk. The equality constraints are:
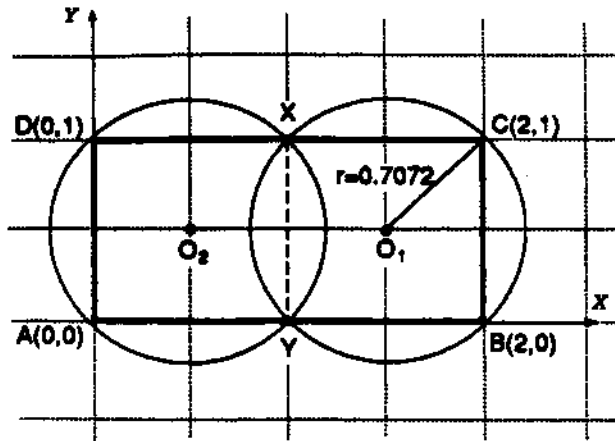
$$X: \qquad \text{if } e = 1$$

8

Figure 6   Optimal cut of a rectangular on AB and CD

$$Y: \quad y_4 = 0$$

The inequality constraints axe:

$$d(C, O_1) \leq r: \quad (x_1 - 2)^2 + (y_1 - \text{I})^2 \leq r^2$$

$$d(B, O_x) < r: \quad (x_1 - 2)^2 + (y_1 - 0)^2 \leq r^2$$

$$d(D, O_2) < r: \quad (x_2 - 0)^2 + (y_2 - \bullet \text{I})^2 \leq r^2$$

$$d(A, O_2) < r: \quad (x_2 - 0)^2 + (y_2 - 0)^2 \leq r^2$$

$$d\{X, O_x) < r: \quad (x_3 - x_l)^2 + (y_3 - y_1)^2 \leq r^2$$

$$d(X, O_2) < r: \quad (x_3 - x_2)^2 + (\text{jft} - y_2)^2 \leq r^2$$

$$d(Y, O_x) < r: \quad (x_4 - x_l)^2 + (y_4 - y_1)^2 \leq r^2$$

$$d(Y, Oi) < r: \quad (x_A - x_2)^2 + (y_4 - V2?)^2 \leq r^2$$

The geometry constraints are:

$$0 < xi < 2$$

$$0 < yx < 2$$

$$0 < x_2 < 2$$

$$0 < y_2 < 2$$

$$0 < x_3 < 2$$

$$0 < y_3 < 2$$

$$0 < x_4 < 2$$

$$0 < y_4 < 2$$

9
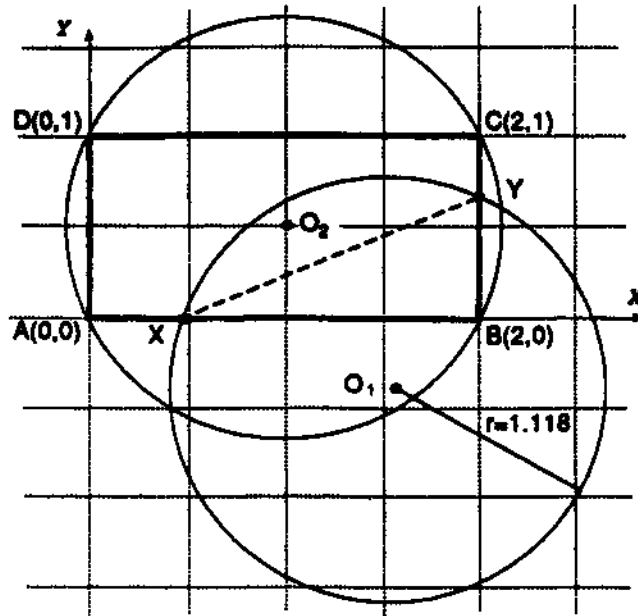
Figure 7   Optimal cut of a rectangular on AB and BC

The computed results using MATRIXx Optimization Module are

$$
\begin{bmatrix}
r \\
x\backslash \\
J\dot{\imath} \\
x_2 \\
92 \\
x_3 \\
J\!e \\
xA \\
y^*
\end{bmatrix}
=
\begin{bmatrix}
1.8452 \\
1.7761 \\
1.4998 \\
3.9255 \\
1.5000 \\
2.8510 \\
3.0000 \\
2.8509 \\
0.0000
\end{bmatrix}
$$

The history of **the** radius r is as follows

r=[2.5   3.0122   2.68   2.5781   2.4835   2.3915   2.3013   2.2163   2.1314
    2.049   1.9696   1.8939   1.8403   1.8452   1.8452   1.8452]

The results of the optimization are shown in Figure 6. If we choose different edge pairs, the resultant optimal radii are different. This can be seen from Figure 7, when CB and AB are selected as an edge pair. The optimum value of each variable is as follows.

10

$$\begin{bmatrix} r \\ x_1 \\ y| \\ X\text{-}l \\ V7 \\ *3 \\ \text{lft} \\ *4 \\ VA \end{bmatrix} \mathbf{s} \begin{bmatrix} 2.1246 \\ 1.5834 \\ 1.5834 \\ 3.4943 \\ 3.5011 \\ 1.9325 \\ 3.7730 \\ 3.7730 \\ 1.9325 \end{bmatrix}$$

The history of the optimized radius r is

r=[2.5 2.9662 2.5370 2.3821 2.2887 2.21955 2.1725 2.1433
   2.1276 2.1226 2.1246 2.1246]

The next section derives a sufficient condition for a local optima to be a global one, giving us a means of getting the global optimum without enumerating all local maximas.

3.2 Searching for a Global Optimum

In the previous section, we described a formulation to obtain an optimal cut passing through a given edge pair. This results in a locally optimal cut - local over the given edge pair. A simple way of obtaining a global optimum over the whole polygon would be to generate local optimas over all possible edge pairs. The edges that produce the globally optimal cut are referred to as *optimal edges.* This however is computationally very expensive. In this section we determine a sufficient condition for a locally optimal cut to be the globally optimum one. We present a simple but effective method for selecting edge pairs such that the globally optimum solution could be obtained without much search.

Consider the problem of selecting two edges as a first edge pair on which the local optimization should be performed. We arrange edges of a polygon in descending order of their lengths $(\pounds i, \pounds 2, \ldots, \pounds_n)$, $L_x \geq L2 \geq \ldots \geq L_n$, where *n* is the number of edges of the polygon. The two longest edges are selected **first.** Further selection is based on the results obtained from the locally optimal solution for this edge pair. Based on the computed radius r_x with respect to L_x and L_2, three possible **cases** may occur.

Case (1):
$$2r_x \leq L_x \text{ and } 2r_x \leq L_2 \tag{27}$$

In this case, the globally optimal solution has been found , which is proved as follows by contradiction. Let us assume that L_x is not one of the optimal edges. For the globally optimal solution (say with radius r_{op}), L_x is fully contained within one of the disks. The longest edge that an optimal disk may contain is of length $2r_{op}$. Therefore,

$$2r_{ov} \geq L_x \tag{28}$$

However, since the radius r_{op} is an optimal solution, it should be less than any possible radius including r_{Xj} i.e.,

$$2r_{op} \leq 2r_1 \tag{29}$$

11

Thus, from (27) we have

$$2\text{rop} \leq 2r_x \leq L_x \tag{30}$$

Unless all equalities are satisfied, (30) is contradictory to (28). Therefore our assumption that L_x was not the optimal edge was wrong. L_1 must be the optimal edge. In the same manner, we can show that $£2$ is the optimal edge too. Since there are only two optimal edges for a convex polygon, and we have identified them and the corresponding radius r_1 is a global optimum.

It must be noted that the condition (27) is a sufficient condition for a global optimal solution, but not a necessary one, i.e. an edge that does not satisfy this condition may also be an optimal one.

Case (2):

$$L_2 \leq 2r_x \leq L_x \tag{31}$$

In this case, we can determine that L_1 is an optimal edge in the same manner as discussed in case (1), but we can not identify whether L_2 is the optimal edge. However, since we have identified L_1 as one of two optimal edges, we can use L_1 as one edge and any other edge I_i ($i = 3,...,n$) as the other to perform optimization. A global optimum can be determined by taking the smallest r_t.

A question is whether the sufficient condition similar to (27), i.e.,

$$2r_i \leq l_1 \quad and \quad 2r_i \leq £_i \tag{32}$$

would become true, from the local optimization result, since in this case we could use it to identify the second optimal edge as we did previously. Unfortunately, (32) cannot occur. For any edge pair other than $£2^*$ the resultant disk should be large enough to contain L_2. Thus

$$2r_t \geq L_2 \tag{33}$$

but

$$L_2 \geq Li \tag{34}$$

therefore,

$$2r_t \geq Li \tag{35}$$

which is contradictory to (32). Thus it becomes necessary for us to examine each case, i.e., for L_i ($t = 3,...,n$), to obtain the globally optimal solution.

Case (S):

$$2r_t \geq L_x \quad and \quad 2r_x \geq L_2 \tag{36}$$

In this case, we can not identify either edge as the optimal edge. Also, because of the same reason discussed in case 2, it is impossible to satisfy the sufficient condition similar to (27) from the result of each case. Therefore, we have to examine each possible edge pair, i.e., for, $i = 1,...,n - 1$, $j = i + 1,...,n$, perform optimization, and compare the resultant radius to finally obtain the global optimum.

However, when we perform local optimization for each case, the following condition may hold true.

$$Li \leq 2r_{ti} \leq Lj \tag{37}$$

where r_y is the resultant radius of the local optimization using edges Z_i and Lj. In the case of (37), we may identify Lj as being an optimal edge.
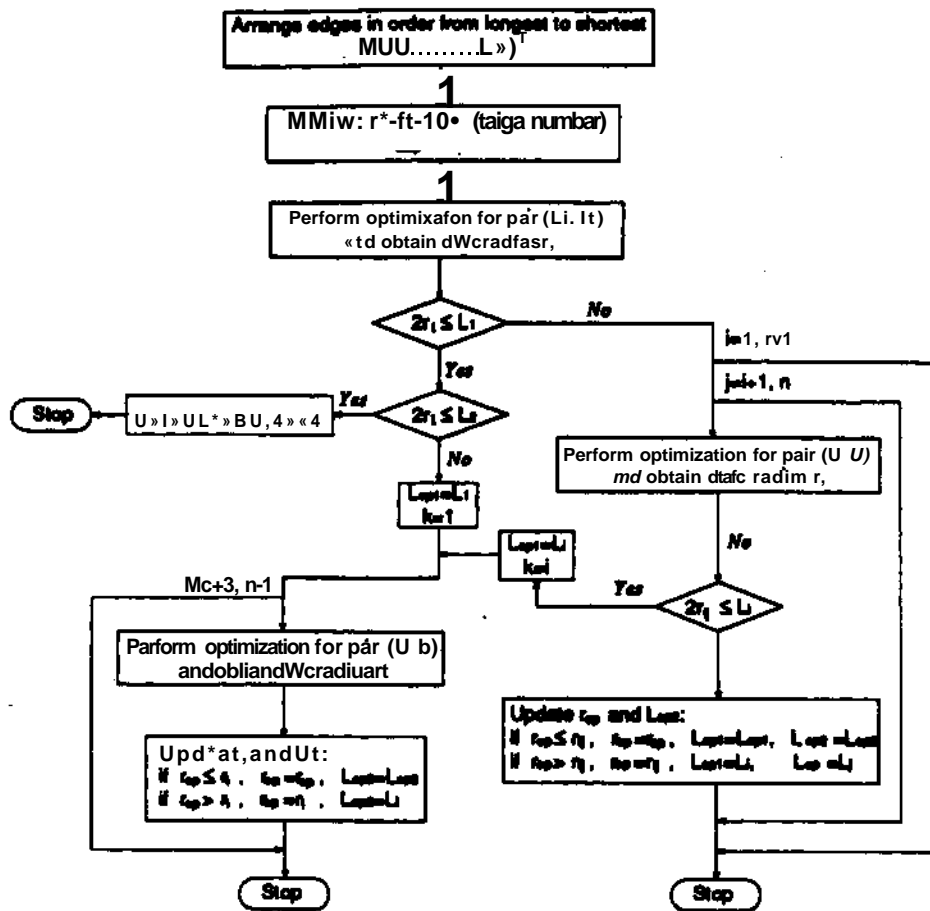
Figure 8   Diagram of the algorithm for generating two minimal disks for a convex polygon

Once we have identified *one* optimal edge, the procedure to find the other is similar to case (2). Based on above discussion, a block diagram for determining a global optimum is shown in Figure 8.

Simulations have been done for various polygons using the proposed algorithm. Two examples are given in Figures 9 and 10.

3.3   Extension to Concave Polygons

The discussion so far is limited to the convex polygon case. In this section, we first discuss the extension of the method to a concave polygon case, which causes an additional amount of computation. Then, we discuss the case when a concave polygon is approximated as a convex polygon.

Let us investigate the difference between cutting a convex polygon and a concave one. For a convex polygon, given a certain edge pair to be cut for optimization, vertices of polygon can be grouped into two, the one enclosed within the *left disk,* and the other in the *right disk.* Given the two edges, this division is always clear. For a concave polygon, however, this division is not clear. For example, a polygon shown in Figure 11 has two concave vertices, D and F. If we use the two longest edges (AC, BC) as the first edge pair to perform local optimization, we have a problem of determining whether D and F should be enclosed within the left disk or the right disk. An arbitrary
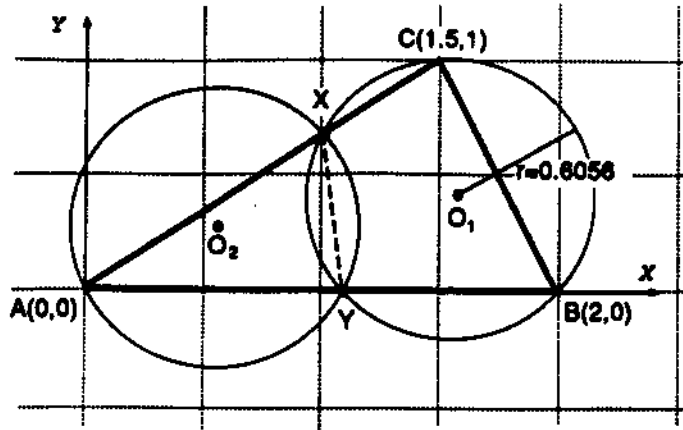
13

Figure 9   Generating two minimal disks for covering a triangle

division does not guarantee the correct solution. When D and F are considered in the left disk, the two disks generated are shown in Figure 11, while when they are considered in the right disk, the two disks generated are shown in Figure 12.

Of course, we could generate all possible divisions and find the local optimum for the two edges. For TO vertices that are not determined, we must perform local optimization JV times, and

$$N = \sum_{Jfe=0}^{m} C_m^k \tag{38}$$

where

$$C_m^{ik} = \frac{m!}{(m - k)!k!} \tag{39}$$

We now suggest two ways of potentially reducing the computational cost. In the first method, we calculate the distance from the indeterminate vertices to the two edges on which we are performing local optimization. The vertices are grouped into two depending on their closeness to one edge over the other. Referring to Figure 13, the vertices Vi, V_2 and V3 can be grouped into two, those close to the edge AB (Vi and *V3)* and those close to the edge CD *(V2)*. Then, we calculate the distance from those points to the two ends of the corresponding edge; $V\backslash$ is close to the end A, V_3 is close to the end B, and Vj is dose to the end D. Generally we have sufficient reason to claim that, the vertex $V\backslash$ should be considered in the left part (i.e., with the ends A and C), while vertices V_2 and V3 should be considered in the right part (i.e., with the ends B and D).

As an alternative, we may first try the cut in such a way that those edges that are adjacent to the undetermined vertices are broken. Then for the result of the local optimization using these edges, we may be able to exclude all other cases. The rationale can be explained using an example in Figure 14. We cut the polygon such that the edges adjacent to the undetermined vertices V_x and *V2* are broken (i.e., the edge V_xA, ViC, V_2B, V2D), and assume the resultant radius is *TABCD-* Then, if

$$2r_{ABCD} \le rn^*x(AD, ED, EF, BF) \tag{40}$$

the result is an optimal solution. This is because when these edges are not broken, i.e., V_x is included within the disk of the ends A and C, and V_2 are included within the disks of the ends
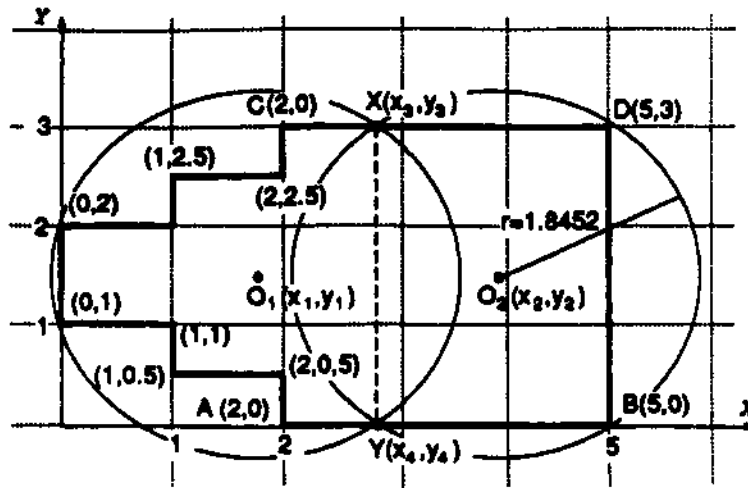
14

Figure 10   Generatln two minimal disks for covering a polygon

B and D, the resultant diameter of the disk must be at least greater than the maximal length of these edges. Therefore, the resultant radius should be greater than *TABCD** and thus $r^\wedge BCD$ is the optimal solution. Of course, if the above equation is not satisfied, a further try must be done, and the similar analysis can be used to identify the optimal solution.

For the polygon in Figure 11, if we consider a cut such that D and F are within the right disk, i.e., AD, DE, EF, and FB are broken, the optimum radius of the disk r is 2.1213 and the resultant disks are shown in Figure 12. This result implies two meaningful points. First, because 2r < *AC* and 2r < *BC,* the edge pair (AC, BC) is the optimal edge pair. Second, since r < *jmx(AD₉EDjEFᵧBF)y* D and F should be enclosed in the right disk. Based on these two conclusions, we claim that r is a global optimal solution.

Since an additional computation is needed to obtain an optimal solution for the minimal disks covering a concave polygon, it is meaningful to investigate what happens if a concave polygon is approximated as a convex one, and the proposed method is used to find an optimal radius of the two disks for the resultant convex polygon.

For an arbitrary concave polygon, it is always possible to connect certain vertices and thus determine a *hypothetical convex polygon,* as shown in Figure 15. The way of determining the hypothetical convex polygon is suggested as follows. Connect each pair of vertices, and examine whether the connecting line lies fully in the outside of the polygon. If it is true, we use the connecting line to replace the edge(s) which locate within the new polygon generated by the connecting line and other edges of the original polygon, i.e., the connecting line *P1P2* replaces the edges Pi*A, AB, BCj CP2,* in Figure 15. If it is not true, skip it. Then, the same process is done for a resultant polygon generated by the previous process, till for any pair of vertices the connecting line is not fully in the outside of the polygon. It is obvious that the generated hypothetical convex polygon is the *minimal convex polygon* that contains the original concave polygon.

For the generated hypothetical convex polygon, we always can use the proposed method to obtain the radii of the minimal disks which are denoted by $r^\wedge_{yfVex}$. We suppose the radii of the minimal disks for the original concave polygon are r_{cave}. What we would like to know is the difference between the $r^\wedge_{yit;ex}$, and r_{cave}, so that the effect of the approximation of a concave polygon using the hypothetical convex polygon becomes clear. Since the area of the concave polygon is a
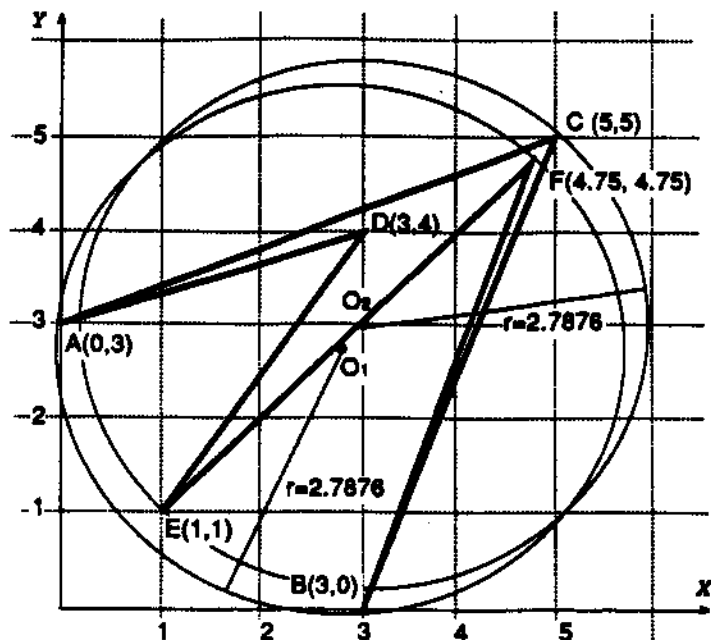
15

Figure 11   Concave polygon case

subset of the area of the generated hypothetical convex polygon, (see Figure 15), the radii of the disks that cover these two polygons have the following relation.

$$r_{cave} \; \leq \; r_{hy,vex} \tag{41}$$

This means when the hypothetical convex polygon is used to approximate the concave polygon, the result is conservative. Now we would like to know how conservative this approximation is. In other words, it is desirable to have an upper limit of the $r^{\wedge}_{VIVCX}$ with respect to r_{cave}. In what follows, we prove that the maximum $r^{\wedge}_{ves}$ is $l.25 r_{cave}$.

Referring to Figure 16, without lose of generality, we assume the radii of the minimal disks for a concave polygon are unity. The hypothetical convex polygon is generated by connecting the vertices A and B, and the resultant convex polygon generally cannot be fully covered by these two disks. However, if we create two lines along the tangent of the two disks, *P1P2,* and P3P4, connecting to the two disks, the area of the ringlike boundary *P1P2D1P3P4D2* surely contains the hypothetical convex polygon. This is because all vertices are enclosed in the two disks, any connecting line between two vertices should be within the area of *P1P2D1P3P4D2.*

Next, we will determine two disks that fully cover the hypothetical convex polygon. Because the ringlike area covers the convex polygon, if we can create two disks that cover the ringlike area, the convex polygon is surely convered by the two disks. The original two disks (r_{cave}, $O_o|$) and (r_{C}ave, $Oo2$) intersect at $F|$ and F_2, and the line $F1F2$ intersects with the lines P_XP_2 and *P3P4* at $S|$ and 5_2, respectively, see Figure 17. Then, we draw a perpendicular bisector of the line $S|$ and 5_2, which intersects with the boundary of the disks at $7|$ and T_2. Fitting each set of three points ($\$i,52,2i$), and ($5i,52,r_2$), we create two disks shown as dashed line in Figure 17. Since the two disks generated contain the ringlike area *P1P2D1P3P4D2* which contains the hypothetical convex polygon, we may claim that the hypothetical convex polygon is contained by these two disks which are denoted by (r&, *Obi)* and (r&, $0*2$), (see Figure 18).
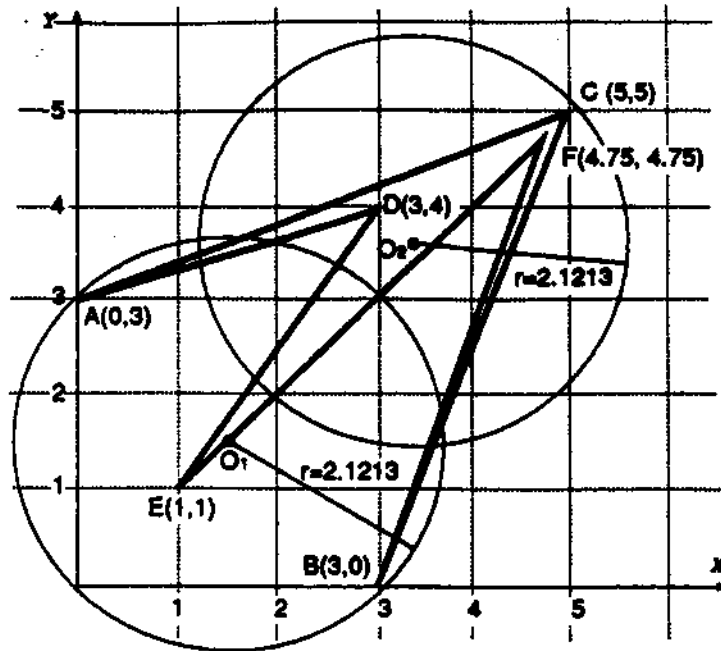
16

Figure 12 Concave polygon cat*

Let x be the distance from the center of the original disks with the radii $r^{\wedge}ve$, $O_o|$, to the center of the generated disks Obi, and $2a$ be the distance between two centers of the original disks O_{oi} O_o2J as shown in Figure 18. The radius r_h can be determined by

$$r_b^2 = (a - x)^2 + 1^2 \tag{42}$$

but,

$$r_b = 1 + x \tag{43}$$

thus,

$$(1 + x)^2 = (a - x)^2 + 1^2 \tag{44}$$

The solution of the above equation is

$$x = \frac{a^2}{2 + 2a} \tag{45}$$

From (43), the **radios** n is

$$r_b = 1 + x = \frac{a^2}{2 + 2a} + 1 \tag{46}$$

It is noted that the maximum of a is unity, when the two disks intersect at one point, the tac-point, and the minimum of a is zero, when the two disks become one. Also, the derivative of function r_b is non-negative, and thus the radius function rj is monotonic increasing, as $a \in [0,1]$. The maximum rj, can be determined as $a = 1$,

$$(rb)max = 1.25 \tag{47}$$

The above derivation is based on a unity r_{Mtie}, and when r_{cave} is not unity we have
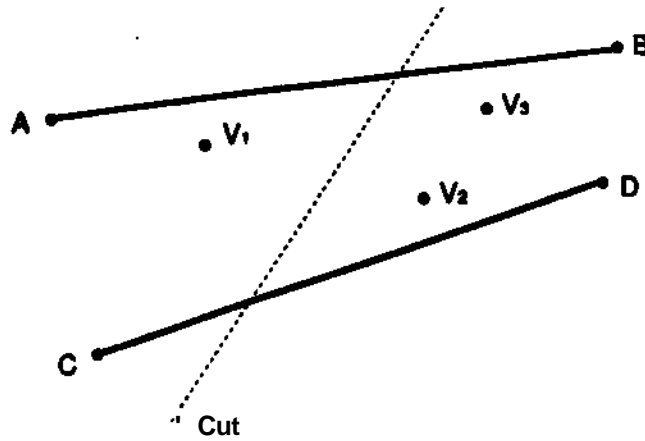
$$r_b \leq 1.25r_{cave} \tag{48}$$

17

Figure 13 Vertices are grouped into two depending on the distance to the edges

We may note that the two disks created by each set of three points, (5i,S2,7i), and (Si,S2 J'_2), may not be the minimal disks for the hypothetical convex polygon, but the minimal disks is definitely not greater than these disks by the definition, i.e.,

$$^r hy_, vex \quad \wedge \quad ^T b \tag{49}$$

Consider (48), we have

$$r_{hy,vex} \leq r_b \leq 1.25 r_{cave} \tag{50}$$

or,

$$r_{cave} \geq 0.8 r_{hy,vex} \tag{51}$$

Combining (41) with (51), we finally obtain

$$0.8 r_{hy,vex} \text{ S } \wedge cave \text{ S } ^r hy_, vex \tag{52}$$

Thus, we may have a conclusion as Theorem 3.

> **Theorem 3:** *If a concave polygon is approximated by a convex polygon, the maximal radius of **the disk** covering the resultant convex polygon is 1.25 times of the radius of the disk that covers the original concave polygon.*

Theorem 3 indicates that using the hypothetical convex polygon from a concave polygon, the optimal solution for the two minimal disks covering the given concave polygon would not be too conservative.

4 Conclusions

We have studied the problem of planning the motion of a planar object in the free space between obstacles. Using a path on the medial axis, we propose the two-disk motion planning strategy to navigate the object from the initial position to the final position subjected to the given constraints.
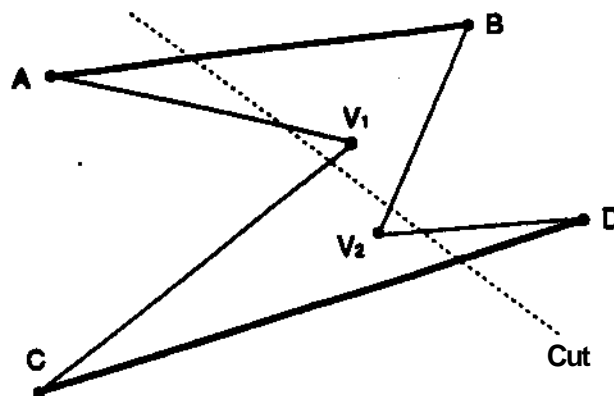
18

**Figure 14   Trying to cut the polygon such that the edges
adjacent to the undetermined vertices are broken**

We at first determine two minimal overlapping disks that fully cover the polygonal object, and then constrain the centers of two disks to move continuously along the path on the medial axis.

In this paper, we focus our effort to determine the two minimal overlapping disks for a given arbitrary polygon. We have proved that the optimal two disks have equal diameters for any polygon. Based on the geometry of the polygon, we formulated the problem as an optimization problem to determine a local optimum for a given edge pair. An algorithm to determine a global optimum from the local optimization results is proposed.

Extension of the method to the concave polygon results in an additional amount of computation. Two intuitive ways to potentially reduce the computation were presented. We propose to approximate the concave polygon by a hypothetical convex polygon. We have proved that the maximal radius of the disk covering the resultant convex polygon is 1.25 times of the radius of the disk that covers the original concave polygon.

Simulations were presented for a variety of polygons. The method is being used for disassembly motion planning of a subassembly within its parent subassembly. This method for path planning in 2-D is feasible for moving objects whose aspect ratio is in the neighborhood of 2. If the object is fairly slim, the given solution is conservative.
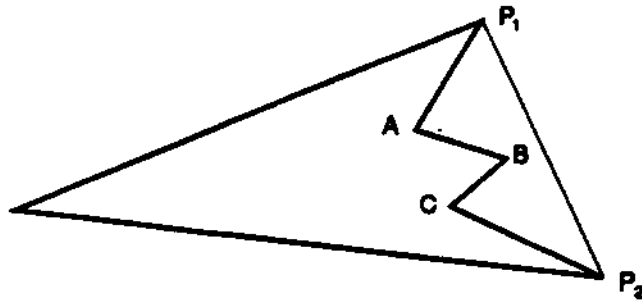
Figure 15   A concave polygon can be approximated by a hypothetical convex polygon.
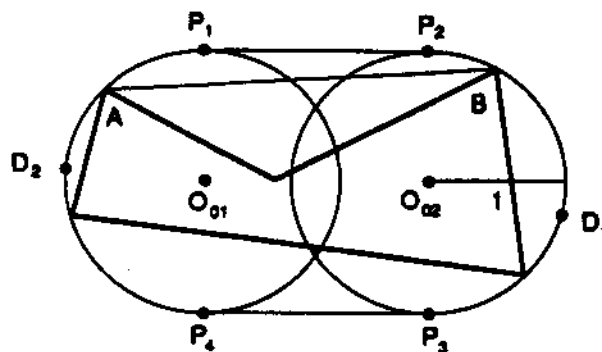

Figure 16   The optimal two disks with unit radii cover a given concave polygon. The hypothetical convex polygon generated from the concave polygon normally is not covered by two disk, but is covered by a ringlike area P1P2D1P3P4D2.

References

[1] R.A. Brooks. Solving the find-path problem by good representation of free space. *IEEE trans. on Systems, Man and Cybernetics,* SMC-13, 1983.

[2] J. Canny. *The complexity of robot motion planning.* The MIT Press, 1988.

[3] M. Sharir J. Schwartz and J. Hopcroft. *Planning, geometry, and complexity of robot motion.* Ablex Publishing Corporation, 1986.

[4] D.G. Kirkpatrick. Efficient computation of continuous skeletons. In *Proceedings of 20th IEEE Annual Symp. on Foundations of Comput. Science,* pages 18-27, 1979.

[5] D.T. Lee. Medial xis transformation of a planar shape. *IEEE Trans, on Patten Analysis and Machine Intelligence,* 4:363-369, 1982.
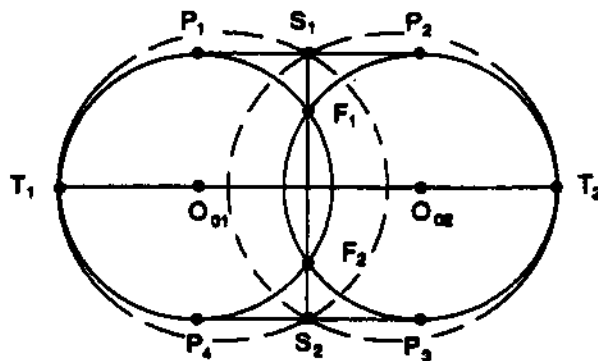
Figure 17 The tangent lines of the disks PiPa and P3P4 intersect with the line F1F2 at Si and Sa. The bisector line of SiSa intersects with the disks at Ti and Ta. Fitting SiSaTi and S1S2T2 determines two larger disks, respectively.
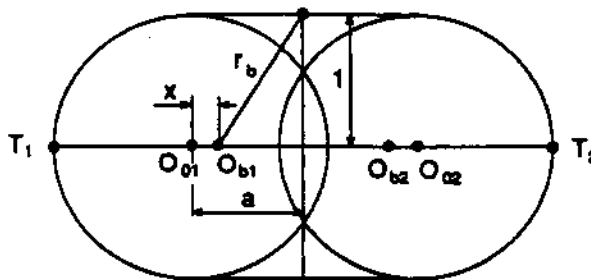

Figure 18 The distance between the center of the larger disk Obi and the center of the original disk Ooi is denated by x, and the distance between two centers of the original disks is denoted by 2a.

[6] T. **Lozano-Perez. Automatic planning of** manipulator transfer movements. *IEEE Trans, on Systems, Man and Cybernetics SMC,* **11:681-698,1981.**

[7] **T. Lozano-Perez. Spatial planning:** A configuration space approach. *IEEE Trans, on Computers,* C-32:108-120,1983.

[8] R.O.Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, New York, 1973.

[9] P.V. Souza and P. Houghton. Computer location of medial axes. *Computers and Biomedical Research,* 10:333-343, 1977.

[10] Y. Xu, R.S. Mattikalli, and P.K. Khosla. Generation of patial medial axis for disassembly motion plann. *(unpublished),* 1991.