

DETERMINING IN LINEAR TIME THE
MINIMAL AREA CONVEX HULL OF TWO
CONVEX POLYGONS UNDER TRANSLATION

Hyun-Chan Lee
Tony C. Woo

The University of Michigan
Dept. of Industrial & Operations Engineering

Tech. Report #86-3

**Determining in Linear Time the Minimal Area Convex
Hull of Two Convex Polygons Under Translation**

Hyun-Chan Lee
Tony C. Woo

The University of Michigan
Department of Industrial & Operations Engineering
Ann Arbor, Michigan 48109

January 1986

<Abbreviated Title>

Minimal Convex Hull of Two Convex Polygons

<Mailing Address>

Tony C. Woo
Department of Industrial & Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109

Abstract

Given two non-overlapping convex polygons P and Q , we find their relative position such that the convex hull encasing them is minimal in area. Without loss of generality, we allow Q to translate about a fixed P . Let N be the total number of vertices in P and Q . We are able to determine the minimal area convex hull in $O(N)$ time. Instead of computing the convex hull after every translation, we compute the slope of the added area function and update it in constant time. We show that the added area function is piecewise linear and that there are only $O(N)$ points on the curve where updating is necessary.

List of Symbols

A_c	:	Added area of the current configuration.
A_{min}	:	Minimal added area.
at_{xc}	:	Adjacent refraction point in $\{t_{xc}\}$.
C_b	:	Sum of constants corresponding to the base edges in the slope formula of the added area function.
CHV	:	Convex hull vertex.
{CHV}	:	Set of CHV.
C_r	:	Sum of constants corresponding to the current reference edge in the slope formula of the added area function.
e_c	:	Current reference edge.
e_i	:	Initial reference edge.
me_c	:	e_c when the added area is minimal.
mv_c	:	v_c when the added area is minimal.
mt_c	:	t_c when the added area is minimal.
n	:	Total number of vertices of polygon P.
N	:	Total number of vertices of polygons P and Q.
P, Q	:	Convex polygons.
S_i	:	Slope of the edge $v_i v_{i+1}$.
S_β	:	Slope of the reference edge e_c ($v_a v_\beta$).
t_c	:	X component of the translated distance along e_c from the vertex v_β to the current refraction point.
$t_x(t_y)$:	X(Y) component of the translated distance along e_c from the current refraction point.
t_{xc}	:	X component of the translated distance along e_c from the

current refraction point to the next refraction point due to a change in the convex hull vertex.

- $\{t_{xc}\}$: Set of t_{xc} .
- t_{xr} : X component of the translated distance along e_c from the current refraction point to the vertex v_a .
- v_i : i^{th} vertex of P and Q for $i=1,2,\dots,N$.
- $v_{pa}(v_{qa})$: Convex hull vertex belonging to P(Q) in the front added polygon.
- $v_{p\beta}(v_{q\beta})$: Convex hull vertex belonging to P(Q) in the rear added polygon.
- $v_{pc}(v_{qc})$: Current vertex of P(Q).
- $v_{pr}(v_{qr})$: Initial touching vertex in P(Q).
- $v_a(v_\beta)$: Vertex of the reference edge e_c in the front (rear) added polygon.
- v_c : Current reference vertex.
- $X_i(Y_i)$: X and Y coordinates of v_i .

List of Figures

- Figure 1. Added Area Function
- Figure 2. Convex Hull of Two Polygons
- Figure 3. Special Case Configurations of Convex Hull
- Figure 4. Base Edges of Triangles in the Added Polygons
- Figure 5. Notations for Convex Hull
- Figure 6. Calculation of the Area of a Triangle
- Figure 7. Changes of a Convex Hull Vertex
- Figure 8. Special Case : More than One Convex Hull Vertex Changes
- Figure 9. Candidate Angles of Rotation
- Figure 10. Reference Edge and Vertex
- Figure 11. Refraction Point Due to a Change in the Convex Hull Vertex
- Figure 12. Finding New Reference Edge and Vertex

List of Procedures

Procedure Find_e_v_c (e_c, v_c, v_{pc}, v_{qc})

Procedure Find_CHV_A_c ($\{CHV\}, A_c, v_{pr}, v_{qr}$)

Procedure Slope_of_added_area_function ($\{CHV\}, C_b, C_r$)

Procedure Calculate_t_xc ($CHV, \{CHV\}, S_\beta, t_{xr}, \{t_{xc}\}$)

Procedure Update_slope_CHV ($\{CHV\}, at_q d(xc), C_b, C_r$)

Algorithm 1. Find_minimum_area_convex_hull

1. Introduction

In computer science and operations research, allocating finite resources has been an important problem. This problem has many applications in industry and is characterized by the clustering of demands to minimize waste (or to maximize utility). One criterion for classifying the problem is the shape of the material to use. If the shape is linear, planar, or spatial, then the problem is considered a one-dimensional, two-dimensional, or three-dimensional problem, respectively. We shall restrict our attention to two-dimensional problems.

An interesting two-dimensional problem is the packing of shapes. The shapes can be rectangular or non-rectangular. The packing of rectangular shapes is formulated as the Bin-Packing problem [6] and the Cutting-Stock problem [4]. It is NP-hard to determine optimal packing for rectangular shapes [2]. For non-rectangular shapes, it is reasonable to expect even greater difficulties. Recent solutions for rectangular shapes have employed heuristics [2,7]. For non-rectangular shapes, very few papers are published. One approach to simplify the packing of non-rectangular shapes is preprocessing them by circumscribing with rectangles [1] and packing the resulting rectangles [3]. However, this preprocessing produces more waste than packing with the original shapes. Therefore, we try to develop more efficient preprocessing algorithm by clustering the original shapes by two, especially two convex polygons.

In this paper, we consider the problem of packing two convex polygons of given orientations under translation such that the convex

hull over the two polygons is minimal in area. If a polygon is non-convex, we can apply a linear time algorithm to find its convex hull [5]. To reduce the waste, we seek the minimal area convex hull of two convex polygons P and Q with Q being allowed to translate along the boundary of P . Since Q is allowed to translate, there are at least $O(N)$ possible configurations, where N is the total number of vertices of P and Q . For each configuration of P and Q , their convex hull can be found in $O(N)$ time [8]. Hence, brute force leads to an algorithm that would run in at least $O(N^2)$ time.

Instead of calculating the total area of the convex hull of two convex polygons, we find an added area due to the convex hull since the area of two given polygons is constant. We show that an added area function is piecewise linear as shown in Figure 1.1. Exploiting the piecewise linearity of the added area function, we find the minimal added area configuration by calculating the added area at points where the slope of the added area function changes. We call these points refraction points. To calculate the added area at refraction points, we triangulate the added area once at the initial configuration and update the slope of the added area function by calculating the amount of translation and the change in triangle area to reach the next refraction point. Since the triangulation is done once in linear time and the updating at each refraction points is done in constant time, we find the minimal area configuration in $O(N)$ time by showing that the total number of refraction points is $O(N)$.

<Insert Figure 1.1>

This paper is organized as follows. In section 2, definitions and

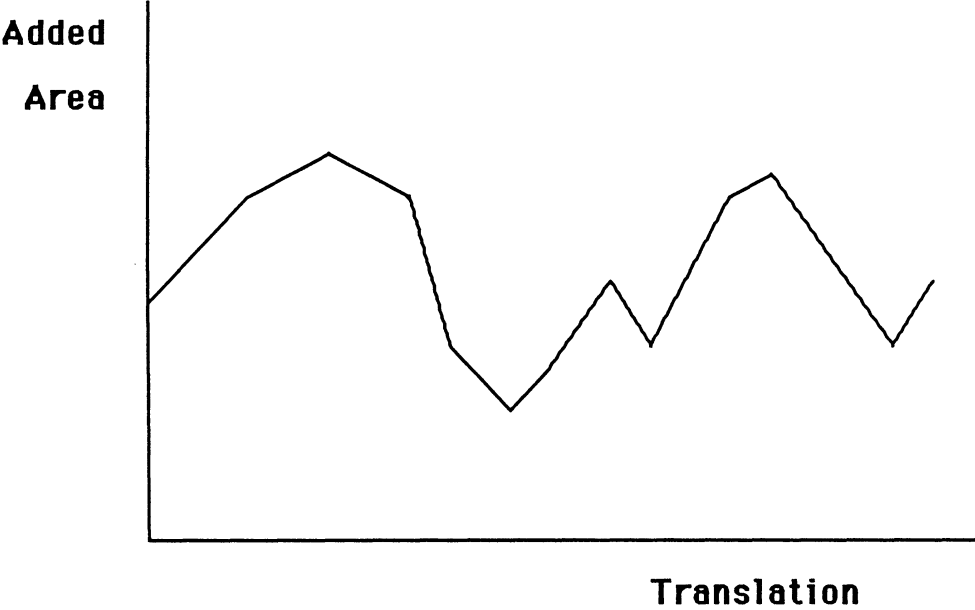


Figure 1.1 Added Area Function

notations are introduced. In section 3, the added area function is examined. In section 4, initialization processes are discussed to find the minimal area convex hull. In section 5, a linear time algorithm and its analysis are presented. In section 6, we conclude this paper.

2. Definitions and Notations

In this section, we introduce the definitions related to the convex hull of convex polygons P and Q, with Q being allowed to translate counter-clockwise along the boundary of P. Then, we introduce the notations for the convex hull and for the vertices and edges of P and Q.

2.1 Definitions

To find the minimal area convex hull of two convex polygons, we define terms we will use including (1) reference edge and vertex, (2) convex hull edges and vertices, (3) front and rear added polygons, (4) triangle and added area functions, and (6) base edge.

(1) Reference edge and vertex : Consider the convex hull over two adjoining convex polygons P and Q as illustrated in Figure 2.1. The two adjoining convex polygons touch each other, with one polygon providing an edge and the other a vertex. The edge, along which Q translates, is called a reference edge. The vertex from the other polygon is called a reference vertex. In Figure 2.1, a reference edge is marked by a double line and a reference vertex by a circle.

<Insert Figure 2.1>

(2) Convex hull edges and vertices : To form a convex hull of two convex polygons, new edges are needed. Each new edge connects two vertices, one vertex from P and the other from Q. The new edges are called convex hull edges and the vertices are convex hull vertices. In general, there are two convex hull edges joining four convex hull vertices. For some cases, the convex hull vertices from P and Q

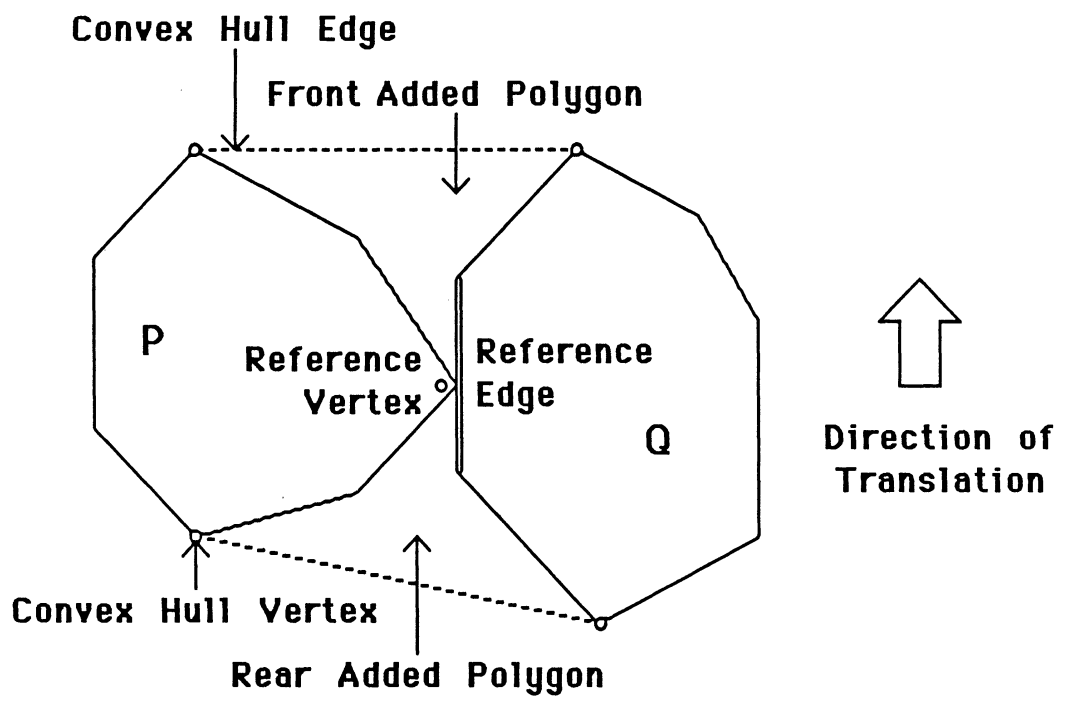


Figure 2.1 Convex Hull of Two Polygons

coincide as illustrated in Figure 2.2. In these cases, we count them as two distinct vertices. Therefore, we can assume that for the general case there are always exactly four convex hull vertices.

<Insert Figure 2.2>

(3) Front and rear added polygons : Once a convex hull of polygons P and Q is formed, there exist two added polygons which are the difference between the convex hull and the polygons P and Q. One of the added polygons is a front added polygon and the other one is a rear added polygon. The front added polygon is the one in the direction Q translates. The area of these two added polygons is an added area. The added area can be zero or positive.

(4) Triangle and added area functions : Finding the minimal area convex hull of two adjoining convex polygons P and Q under translation is equivalent to finding the position of Q with minimal added area. As Q translates along a reference edge, the area of the added polygons changes. Since the added polygons can be triangulated, the change in the added area can be computed from the changes in the area of the triangles. The changes in the area of triangles are described as triangle area functions. The sum of the triangle area functions in the added area is an added area function.

(5) Base edge : In general, a triangle in a triangulated added polygon shares one entire edge with polygon P or Q (see Figure 2.3). The shared edge will be referred to as a base edge. In some cases, triangles share two entire edges with polygons P and Q, one with P and the other with Q. In these cases, selecting the base edge depends on

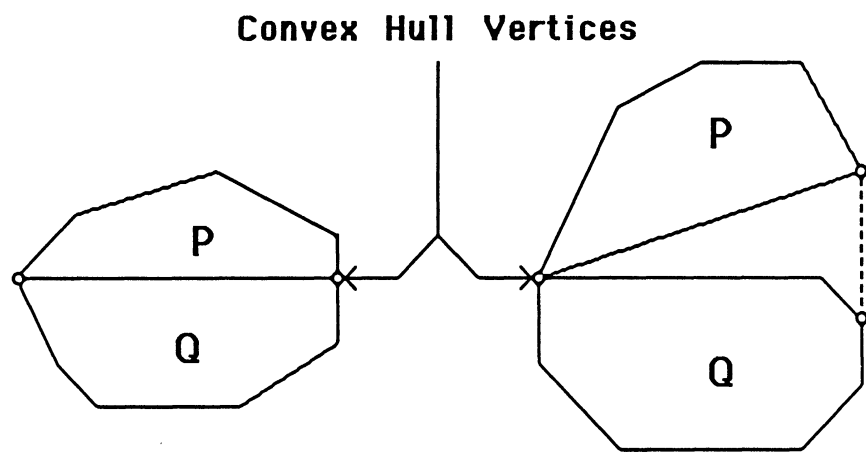


Figure 2.2 Special Case Configurations of Convex Hull

which polygon provides a reference edge. If the reference edge belongs to P, for the triangle in the front (or rear) added polygon, the base edge is the one which belongs to Q (or P). If the reference edge belongs to Q, we select the base edge in the opposite way.

<Insert Figure 2.3>

2.2 Notations

Now we introduce the notations for (1) vertices of polygons P and Q, (2) edges of polygons P and Q, (3) the data structure of vertices, (4) the convex hull vertices, (5) the reference edge, and (6) the reference vertex.

(1) Vertices of polygons P and Q : We assume that the total number of vertices of polygons P and Q is N. Among the N vertices, n vertices belong to polygon P. Hence, Q has N - n vertices. We indicate the i^{th} vertex of polygon P, starting from a fixed vertex and indexing in a counter-clockwise order, as v_i where $i=1,2,\dots,n$. Also, v_i stands for the i^{th} vertex of polygon Q, starting from a fixed vertex and indexing in a clockwise order, where $i=n+1,n+2,\dots,N$. The reason for using the opposite ordering is to simplify the equations of the added area functions. The X and Y coordinates of a vertex v_i are denoted by (X_i, Y_i) .

(2) Edges of polygons P and Q : We indicate an edge by using the notations for vertices. The i^{th} edge of a polygon is $v_i v_{i+1}$ and its slope is S_i .

(3) The data structure of vertices : The vertices of P and Q are

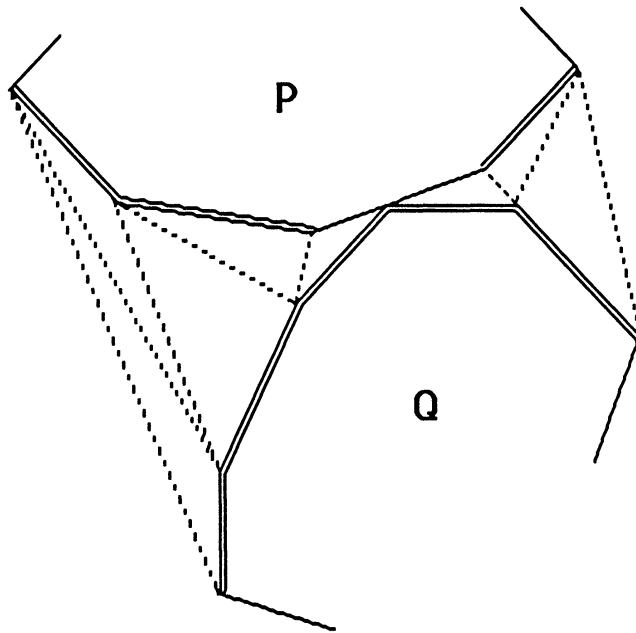


Figure 2.3 Base Edges (double lined) of
Triangles in the Added Polygons

separately stored as doubly linked lists. In the lists, the next (or previous) vertex of a given vertex v_i is v_{i+1} (or v_{i-1}). The next (or previous) vertex is decided in cyclic order, i.e., the next (or previous) vertex of v_n (or v_1) of P is v_1 (or v_n) and the next (or previous) vertex of v_N (or v_{n+1}) of Q is v_{n+1} (or v_N).

(4) The convex hull vertices : To indicate the convex hull vertices, we introduce a special notation. We indicate the vertices as v . If a convex hull vertex belongs to polygon P (or Q), we include p (or q) in the subscript of v and if the vertex belongs to the front (or rear) added polygon, we include a (or β) in the subscript of v . Therefore, the convex hull vertices are denoted by v_{pa} , $v_{p\beta}$, v_{qa} , and $v_{q\beta}$. The locations of these vertices are illustrated in Figure 2.4.

<Insert Figure 2.4>

(5) The reference edge : We classify the reference edge as initial reference edge and current reference edge and indicate these edges as e_i and e_c , respectively. For the current reference edge e_c , we indicate its two vertices as v_a and v_β , where v_a (or v_β) is the vertex belonging to the front (or rear) added polygon. Since v_a is the next vertex of v_β , the slope of the current reference edge e_c is S_β .

(6) The reference vertex : For the reference vertex, we introduce only the current reference vertex and indicate this vertex as v_c .

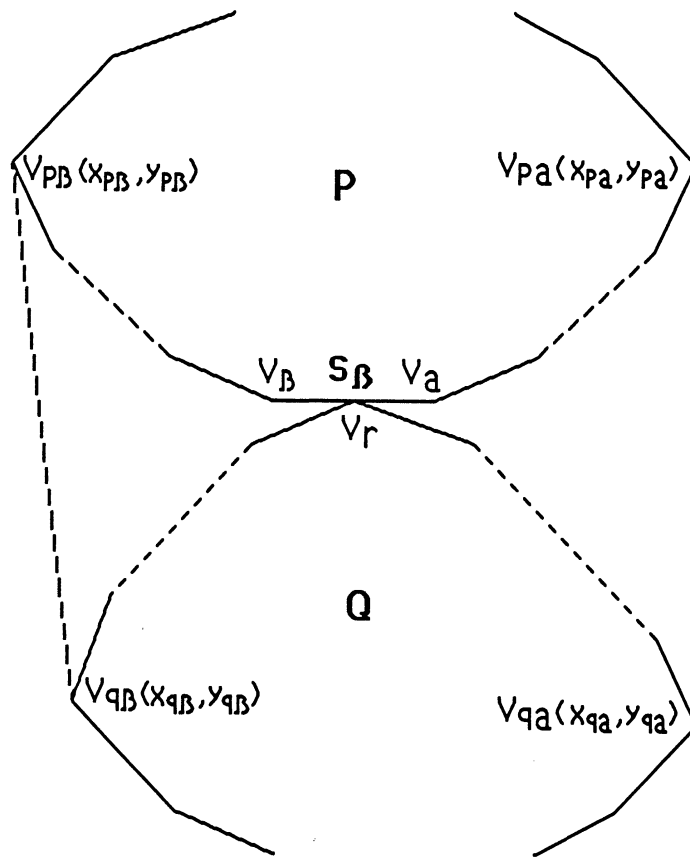


Figure 2.4 Notations for Convex Hull

3. Added Area Function

In this section, we examine the added area function. The added area function is a sum of the triangle area functions since the added area can be triangulated. To characterize the added area function, first we prove that the triangle area function is linear. Based on the linearity, we derive the slope of the added area function. Then, by examining the slope formula, we exploit the conditions when the slope of the added area function changes. Finally, piecewise linearity of the added area function is discussed.

3.1 Triangle Area Function

As a basis of the added area function, we examine the triangle area function as Q translates counter-clockwise along the boundary of P . First, we define a triangle under consideration. Then, the method to calculate the triangle area is explained. Finally, we characterize the triangle area function of a translated triangle as Q translates along the reference edge.

We consider the area of the translated triangle of a triangle $v_i v_j v_k$ when polygon Q is translated by (t_x, t_y) along the current reference edge $e_c (v_a v_\beta)$ with slope S_β , where t_x (or t_y) is the X (or Y) component of the translated distance. We assume that the base edge of the triangle $v_i v_j v_k$ is $v_i v_j$. Since $v_i v_j$ is the base edge, both vertices v_i and v_j belong to the same polygon P or Q and v_k belongs to the other polygon. We assume v_j is the next vertex of v_i in the same polygon. Therefore, the slope of the base edge $v_i v_j$ is S_i . The translated triangle can be denoted by $v_i v_j v_k'$ if the base edge $v_i v_j$ belongs to

polygon P and denoted by $v_i'v_j'v_k'$ if the base edge belongs to Q.

The area of a triangle $v_i'v_j'v_k'$ is computed by the cross product of its two edges. The cross product is taken using left hand rule. The cross product $(v_i'v_j' \times v_j'v_k')/2$ is the area of a triangle $v_i'v_j'v_k'$ if its base edge belongs to P. On the other hand, if the base edge belongs to Q, then the area is $(v_j'v_i' \times v_i'v_k')/2$.

In the following Lemma 3.1, we assume that no edge has an infinite slope. The details for resolving an infinite slope are discussed in Section 4.

Lemma 3.1 The triangle area function of the translated triangle of $v_i'v_j'v_k'$ along e_c by (t_x, t_y) is linear in t_x . Its slope is:

$$(x_j - x_i)(S_i - S_\beta)/2.$$

The proof is given in Appendix.

Lemma 3.1 can be illustrated as follows. Consider a triangle $v_{n+1}v_{n+2}v_1$ in Figure 3.1. The slope of its triangle area function is:

$$\begin{aligned} & (x_j - x_i)(S_i - S_\beta)/2 \\ = & (x_{n+2} - x_{n+1})(S_{n+1} - S_2)/2 \end{aligned}$$

where S_{n+1} is the slope of the base edge $v_{n+1}v_{n+2}$, and S_2 is the slope of the reference edge v_2v_3 .

<Insert Figure 3.1>

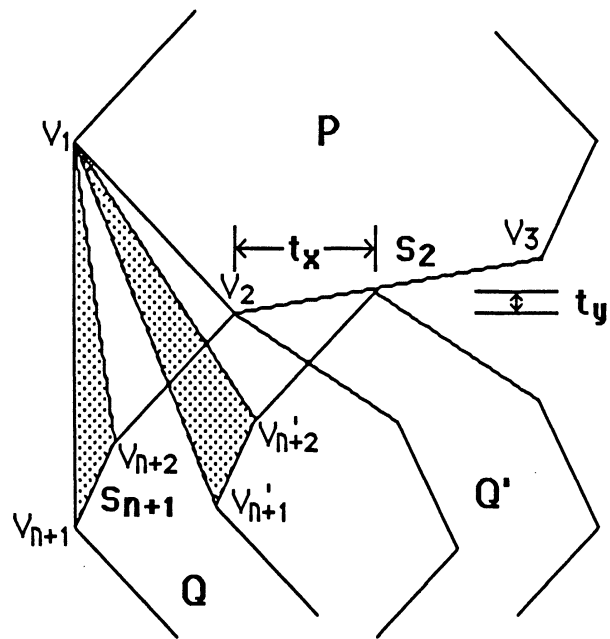


Figure 3.1 Calculation of the Area of a Triangle

As Q translate along the boundary of P , a triangle in the added area polygons exists for certain range of the translated distance of Q . Therefore, a triangle area function is a line segment defined in this range.

3.2 Slope of the Added Area Function

Since the added area is a sum of the triangles in added polygons, we can calculate the added area function from triangle area functions. The added area polygons includes the same triangles for certain range of the translated distance of Q . In the range, the added area function is linear from Lemma 3.1. We calculate the slope of the added area function for a configuration of P and Q .

The slope of the added area function depends on the slope of the base edges and the slope of the reference edge since the slope of a triangle area function depends on the slope of the base edge and the slope of the reference edge from Lemma 3.1. It is worthwhile to note that the slope of the added area function does not depend on how the added area polygons are triangulated. If we know which edges of polygon P and Q correspond to the added polygons and which edge is the reference edge, we can calculate the slope of the added area function.

To calculate the slope of the added area function, we need to know the base edges and the reference edge. From the definition of convex hull vertices, we know that the edges between $v_{p\beta}$ and v_{pa} and between $v_{q\beta}$ and v_{qa} belong to the added polygons and these edges are the base edges, where v_{pa} (or v_{qa}) and $v_{p\beta}$ (or $v_{q\beta}$) are the convex hull vertices belonging to P (or Q) in the front and rear added polygons,

respectively. Therefore, if we assume the current reference edge is e_c ($v_a v_\beta$), the slope of the added area function is:

$$\sum_{i=p\beta}^{pa-1} (x_{i+1} - x_i)(S_i - S_\beta)/2 + \sum_{i=q\beta}^{qa-1} (x_{i+1} - x_i)(S_i - S_\beta)/2$$

where pa , qa , $p\beta$, and $q\beta$ are the indices for the convex hull vertices and S_β is the slope of the current reference edge e_c . Note that if S_i and S_β are the same, $(S_i - S_\beta)$ is zero. This implies that the reference edge is not used as a base edge.

3.3 Change in the Slope of the Added Area Function

Let us examine the change in the slope of the added area function. From the formula in Section 3.2, we can see the slope changes if the number of terms in the summation changes or S_β changes. The former case corresponds to situation when the convex hull vertex changes and the latter case when the reference edge changes. The following two lemmas 3.2 and 3.3 clarify the conditions when the slope of the added area function changes.

Lemma 3.2 The slope of the added area function changes if a convex hull vertex changes.

[proof] If a convex hull vertex changes, the number of terms in the slope formula of the added area function increases by one or decreases by one. In either case, the slope of the added area function changes because the slope of the triangle area function of the appearing (or disappearing) triangle cannot be zero. Because both P and Q are convex polygons, the slope of the base edge cannot

be the same as the slope of the reference edge. Hence, by Lemma 3.1, the slope of the area function of an appearing (or a disappearing) triangle must be nonzero. ■

A change in the convex hull vertex is illustrated in Figure 3.2. In Figure 3.2(a) a triangle $v_{p\beta}v_{p\beta+1}v_{q\beta}$ disappears as a convex hull vertex changes from $v_{p\beta}$ to $v_{p\beta+1}$. This occurs when $v_{q\beta}$ passes the extended line of edge $v_{p\beta}v_{p\beta+1}$. Symmetrically, in Figure 3.2(b), as the extended line of an edge $v_{q\beta}v_{q\beta-1}$ passes the vertex $v_{p\beta}$, a convex hull vertex changes from $v_{q\beta}$ to $v_{q\beta-1}$ and a triangle $v_{p\beta}v_{q\beta}v_{q\beta-1}$ appears.

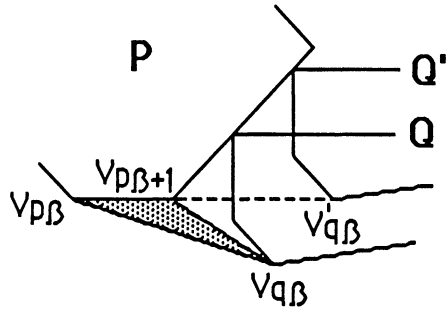
<Insert Figure 3.2>

A special case occurs if more than one convex hull vertex changes simultaneously as illustrated in Figure 3.3. In this special case, the slope of the added area function may not change since occasionally the sum of the adding terms can be same as the sum of the deleting terms. To resolve such a case, we consider the changes of convex hull vertices one at a time. Therefore, in general, the slope of the added area function changes whenever the convex hull vertex changes.

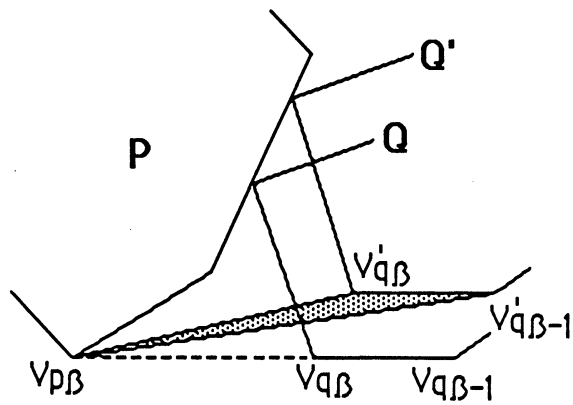
<Insert Figure 3.3>

Lemma 3.3 The slope of the added area function changes if the coefficient of S_β is not zero and the slope of the reference edge changes.

[proof] In the formula of the slope of the added area function, the coefficient of S_β is $(X_{p\beta} - X_{pa}) + (X_{q\beta} - X_{qa})$. This coefficient is not zero by assumption. Therefore, if the value of S_β changes,

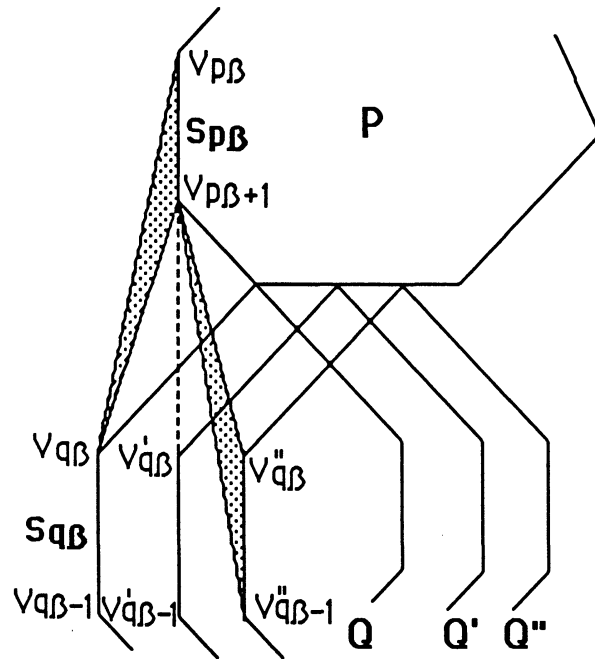


- (a) A convex hull vertex changes from $V_{p\beta}$ to $V_{p\beta+1}$:
A triangle $V_{p\beta} V_{p\beta+1} V_{q\beta}$ disappears.



- (b) A convex hull vertex changes from $V_{q\beta}$ to $V'_{q\beta-1}$:
A triangle $V_{p\beta} V'_{q\beta} V'_{q\beta-1}$ appears.

Figure 3.2 Changes of a Convex Hull Vertex



Two convex hull vertices $V_{p\beta}$ and $V_{q\beta}$ change simultaneously to $V_{p\beta+1}$ and $V_{q\beta-1}$, respectively.

Figure 3.3 Special case:

More than one convex hull vertex changes

i.e. the slope of the reference edge changes, the slope of the added area function changes. ■

3.4 Shape of the Added Area Function

Now we are ready to characterize the shape of the added area function. Lemma 3.2 and Lemma 3.3 imply that the added area function is piecewise linear since there exist points where the slope of the added area function changes. Since the area of a configuration is unique, there can not be a jump in the added area function where the slope changes. Therefore, the added area function is continuous at points where the slope changes. Hence, the added area function is continuous piecewise linear.

Since the area of the polygons P and Q is constant, finding the minimal area convex hull of P and Q is equivalent to finding the minimum in the added area function. Since the added area function is piecewise linear the minimum of the added area function is at one of the refraction points where the slope of the added area function changes. Therefore, by tracing the area at the refraction points, we can find the minimal area convex hull of P and Q . In the following sections, we develop an algorithm to find the minimal convex hull of P and Q in linear time.

4. Initialization

To find the minimal area convex hull, we set the initial values of the parameters to be used in an algorithm in Section 5. First, we set the initial configuration of P and Q. Second, a procedure to determine initial reference edge and reference vertex is presented. Third, we calculate the initial added area and determine the initial convex hull vertices. Finally, we calculate the initial slope of the added area function. All of the above are done in linear time.

4.1 Initial configuration of P and Q

We find the initial configuration of P and Q by translating polygon Q so that a vertex of P and a vertex of Q coincide. In such configuration, there can be edges with an infinite slope. A method to avoid infinite slopes is discussed.

First, we find the initial touching configuration of P and Q. From the coordinates of the vertices of P, we find a vertex v_{pr} with the smallest Y-coordinate. Similarly, we find a vertex v_{qr} with the largest Y-coordinate among the vertices of Q. In case there is more than one such vertices, choose a vertex with the largest (or the smallest) X-coordinate for v_{pr} (or v_{qr}). We translate Q such that v_{qr} coincides with v_{pr} .

In this translated configuration, there can be edges that have an infinite slope. However, there can be at most four such edges, two from P and two from Q, because of the convexity of the two polygons. If both P and Q are rotated by a "small" amount about v_{pr} , then no edge will

have an infinite slope. The angle of rotation is one half of the minimum of the four angles -- θ_i , θ_j , θ_k , and θ_m -- between a vertical line and the four edges which end or start at the vertices of P or Q, respectively, with the smallest and the largest X-coordinates. It is illustrated in Figure 4.1. If there are edges with an infinite slope, the vertices with the largest or the smallest X-coordinate is not unique. In this case, if the vertices belong to P (or Q), we choose the vertex with the smallest (or the largest) Y-coordinate. This is performed in linear time.

<Insert Figure 4.1>

4.2 Initial Reference Vertex and Reference Edge

To translate polygon Q along the boundary of polygon P, there should be an edge and a vertex which provide the direction and position along which Q translates. The edge is the reference edge and the vertex is the reference vertex. We develop a procedure to find the reference edge and the reference vertex for an initial configuration of P and Q.

Initial reference edge and reference vertex may belong to P or Q depending on their relative orientations. The two situations are illustrated in Figure 4.2, where the reference edge is marked by a double line and the reference vertex by a circle.

<Insert Figure 4.2>

To determine the initial reference edge and vertex, and to determine the new reference edge and vertex to be used in Section 5, the Procedure Find_e_c_v_c is provided. The output of the procedure is e_c and v_c , which indicate the determined reference edge and reference vertex,

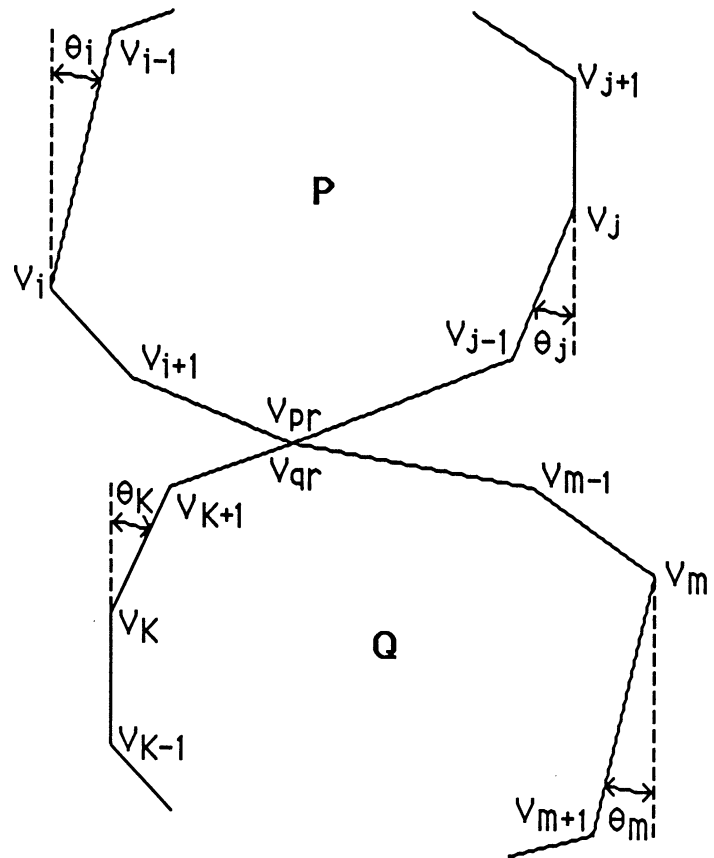
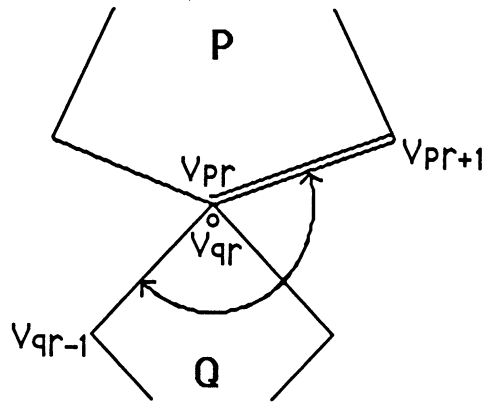
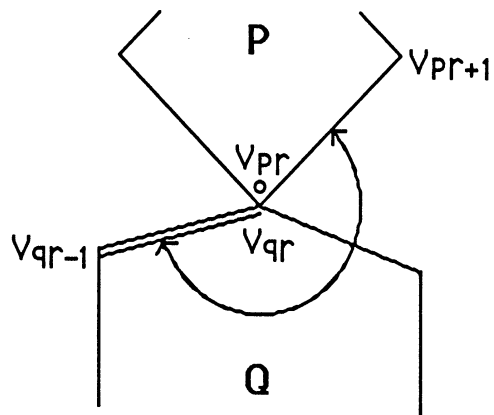


Figure 4.1 Candidate Angles of Rotation



(a) Reference Edge Belongs to Polygon P



(b) Reference Edge Belongs to Polygon Q

Figure 4.2 Reference Edge and Vertex

respectively. The input of the procedure is v_{pc} and v_{qc} , which are two touching vertices from P and Q, respectively. In the procedure, the cross product is taken using the left hand rule.

<Insert Procedure Find_ e_c - v_c >

The procedure works as follows. If the cross product $v_{qc-1}v_{qc} \times v_{pc}v_{pc+1}$ is positive (Figure 4.2(a)), the counter-clockwise angle of $v_{qc-1}v_{qc}v_{pc+1}$ is less than π . In this case, current reference edge e_c is $v_{pc+1}v_{pc}$ which belongs to P and current reference vertex v_c is v_{qc} which belongs to Q. If the cross product is not positive (Figure 4.2(b)), e_c is $v_{qc}v_{qc-1}$ and v_c is v_{pc} .

In the initial configuration, the touching vertices are v_{pr} and v_{qr} . With these vertices as an input, we can determine the initial reference edge and reference vertex by calling the procedure with parameters $(e_i, v_c, v_{pr}, v_{qr})$. The output e_i also represents the current reference edge e_c .

4.3 Initial Convex Hull Vertices and Added Area

To start to find the minimal area convex hull of P and Q, we need to find the initial convex hull vertices and added area. While we find the initial convex hull vertices, we can calculate the initial added area. For these purposes, Procedure Find_CHV_ A_c is provided. In the procedure, a p-cross-product is for the area of a triangle which shares the base edge with P while a q-cross-product is for a triangle which shares the base edge with Q. {CHV} is the set of four convex hull vertices, v_{pa} , $v_{p\beta}$, v_{qa} , and $v_{q\beta}$. A_c stands for the current added area.

Procedure Find_e_c (e_c, v_c, v_{pc}, v_{qc})

begin

 cross_product $\leftarrow v_{qc-1} v_{qc} \times v_{pc} v_{pc+1}$.

if (cross_product > 0)

then $e_c \leftarrow v_{pc+1} v_{pc}, v_c \leftarrow v_{qc}$.

else $e_c \leftarrow v_{qc} v_{qc-1}, v_c \leftarrow v_{pc}$.

end

<Insert Procedure Find_CHV_A_c>

The procedure works as follows. For the front added polygon, starting from the touching vertices v_{pr} and v_{qr} as current vertices of polygon P and Q, respectively, form triangles which share the base edge with Q by traversing the vertices of Q clockwise until no more such triangle can be formed. Move the current vertex of P to the next vertex and test whether a triangle which shares the base edge with P can be formed. If it is possible, form it and restart to form triangles which share the base edge with Q starting from the current vertex of Q. Repeat this process until no more triangle can be formed. Then, put the current vertices of polygons P and Q into the set of convex hull vertices. While forming the triangles, using the cross product, calculate the added area. The same process is applied for the rear added polygon by changing the direction of traversing the vertices.

While performing the Procedure Find_CHV_A_c, every vertex of P and Q is traversed at most once. Therefore, the time complexity of this procedure is linear in N.

4.4 Initial Slope of the Added Area Function

To calculate the initial slope of the added area function, we use the formula in Section 3.2. To simplify the updating the slope, we introduce a representation for the slope. We represent the slope of the added area function as:

$$(C_b + C_r S_\beta)/2 \text{ where}$$

Procedure Find_CHV_A_c ({CHV}, A_c, v_{pr}, v_{qr})

begin

{ checking front added polygon }

v_{pc} ← v_{pr}, v_{qc} ← v_{qr}.

A_c ← 0.

p_cross_product ← v_{qc+1}v_{qc} × v_{pc}v_{pc+1}.

if (p_cross_product > 0) **then** v_{pc} ← v_{pc+1}.

if (p_cross_product = 0) **then**

v_{pc} ← v_{pc+1}, v_{qc} ← v_{qc+1}, p_cross_product ← 1.

while (p_cross_product > 0) **do**

begin

q_cross_product ← v_{qc+1}v_{qc} × v_{qc}v_{pc}.

while (q_cross_product > 0) **do**

begin

A_c ← A_c + q_cross_product/2, v_{qc} ← v_{qc+1}.

q_cross_product ← v_{qc+1}v_{qc} × v_{qc}v_{pc}.

end

p_cross_product ← v_{qc}v_{pc} × v_{pc}v_{pc+1}.

if (p_cross_product > 0) **then**

A_c ← A_c + p_cross_product/2, v_{pc} ← v_{pc+1}.

end

v_{pa} ← v_{pc}, v_{qa} ← v_{qc}.

{CHV} ← {v_{pa}, v_{qa}}.

```

{ checking rear added polygon }

vpc ← vpr, vqc ← vqr.

p_cross_product ← vpc-1vpc × vqcvqc-1.

if (p_cross_product > 0) then vpc ← vpc-1.

while (p_cross_product > 0) do

  begin

    q_cross_product ← vpcvqc × vqcvqc-1.

    while (q_cross_product > 0) do

      begin

        Ac ← Ac + q_cross_product/2, vqc ← vqc-1.

        q_cross_product ← vpcvqc × vqcvqc-1.

      end

      p_cross_product ← vpc-1vpc × vpcvqc.

      if (p_cross_product > 0) then

        Ac ← Ac + p_cross_product/2, vpc ← vpc-1.

      end

    end

  vpβ ← vpc, vqβ ← vqc.

  {CHV} ← {CHV} + {vpβ, vqβ}.

end

```

C_b = Sum of constants corresponding to the base edges

C_r = Sum of constants corresponding to the reference edge

S_β = Slope of the reference edge

then the slope of the added area function shown in Section 3.2 have terms:

$$C_b = \sum_{i=p\beta}^{pa-1} (X_{i+1} - X_i)S_i + \sum_{i=q\beta}^{qa-1} (X_{i+1} - X_i)S_i$$

$$\begin{aligned} C_r &= - \sum_{i=p\beta}^{pa-1} (X_{i+1} - X_i) - \sum_{i=q\beta}^{qa-1} (X_{i+1} - X_i) \\ &= (X_{p\beta} - X_{pa}) + (X_{q\beta} - X_{qa}) \end{aligned}$$

A procedure to calculate the slope of the added area function using above formula is given in the Procedure Slope_of_added_area_function.

<Insert Procedure Slope_of_added_area_function>

Procedure Slope_of_added_area_function($\{CHV\}, C_b, C_r$)

begin

$$C_r \leftarrow (X_{p\beta} - X_{pa}) + (X_{q\beta} - X_{qa})$$

$$C_b \leftarrow 0.$$

$$i \leftarrow v_{p\beta}.$$

while ($i < v_{pa}$) **do**

begin

$$C_b \leftarrow C_b + (X_{i+1} - X_i) S_i.$$

$$i \leftarrow i + 1.$$

end

$$i \leftarrow v_{q\beta}.$$

while ($i < v_{qa}$) **do**

begin

$$C_b \leftarrow C_b + (X_{i+1} - X_i) S_i.$$

$$i \leftarrow i + 1.$$

end

end

5. The Algorithm and Its Analysis

We are ready to present an algorithm that gives the minimal area configuration. The minimal area convex hull for the two convex polygons P and Q corresponds to the configuration in which the added area is minimal. The added area function is piecewise linear and the slope of the function changes at refraction points. To find the minimal area configuration, we trace only the refraction points of the added area function by updating the slope of the function in constant time. We discuss the method to trace the refraction points and to update the slope of the added area function. Now, the algorithm would then run in linear time if there is linear number of refraction points. We show that there can be no more than $3N$ of them where N is the total number of vertices in P and Q .

5.1 Refraction Point Due to Convex Hull Vertex Change

If a convex hull vertex changes, then the slope of the added area function changes as well. There exists a refraction point corresponding to this change. We present a method to calculate the X component of the translated distance of polygon Q along the current reference edge to reach the refraction point. We denote the X component of the translated distance as t_{xc} .

<Insert Figure 5.1>

In Figure 5.1, as Q becomes Q' , the convex hull vertex $v_{q\beta}$ moves to $v'_{q\beta-1}$. The amount of translation t_{xc} can be calculated from the fact that $v_{p\beta}$ is on the extended line of the edge $v'_{q\beta-1}v'_{q\beta}$. The equation of

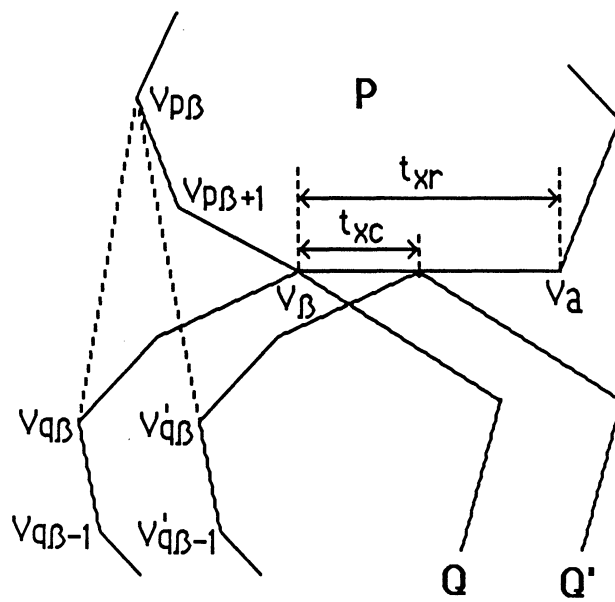


Figure 5.1 Refraction Point Due to a Change in the Convex Hull Vertex

the edge $v'_{q\beta-1}v'_{q\beta}$ is:

$$Y - Y'_{q\beta} = S_{q\beta-1}(X - X'_{q\beta}).$$

By replacing $X'_{q\beta}$ with $(X_{q\beta} + t_{xc})$ and $Y'_{q\beta}$ with $(Y_{q\beta} + S_{\beta}t_{xc})$, we get:

$$Y - (Y_{q\beta} + S_{\beta}t_{xc}) = S_{q\beta-1}(X - (X_{q\beta} + t_{xc})).$$

By inserting the coordinates of $v_{p\beta}$ into the equation, we have:

$$Y_{p\beta} - (Y_{q\beta} + S_{\beta}t_{xc}) = S_{q\beta-1}(X_{p\beta} - (X_{q\beta} + t_{xc})).$$

Then, the refraction point due to the change of $v_{q\beta}$ can be calculated:

$$t_{xc} = (S_{q\beta-1}(X_{p\beta} - X_{q\beta}) - (Y_{p\beta} - Y_{q\beta})) / (S_{q\beta-1} - S_{\beta}).$$

For the general case, we provide the Procedure Calculate $_t_{xc}$. In the procedure, t_{xr} is the X component of the distance Q translates along the current reference edge, and it is explained in the next section.

<Insert Procedure Calculate $_t_{xc}$ >

5.2 Refraction Point Due to Reference Edge Change

From Lemma 3.3, if the reference edge changes, there exists a refraction point corresponding to this change except some special cases. The special cases are the coefficient of S_{β} is zero and S_{β} remains unchanged. We determine the new reference edge and reference vertex when the reference edge changes. Then, we calculate the maximum distance the polygon Q can be translated along the new reference edge.

Procedure Calculate_ t_{xc} **(CHV, {CHV}, S_{β} , t_{xr} , { t_{xc} })**

begin

case CHV **of**

$$v_{pa} : t_{xc} = (S_{pa}(X_{qa} - X_{pa}) - (Y_{qa} - Y_{pa})) / (S_{\beta} - S_{pa}).$$

$$v_{p\beta} : t_{xc} = (S_{p\beta}(X_{q\beta} - X_{p\beta}) - (Y_{q\beta} - Y_{p\beta})) / (S_{\beta} - S_{p\beta}).$$

$$v_{qa} : t_{xc} = (S_{qa-1}(X_{pa} - X_{qa}) - (Y_{pa} - Y_{qa})) / (S_{qa-1} - S_{\beta}).$$

$$v_{q\beta} : t_{xc} = (S_{q\beta-1}(X_{p\beta} - X_{q\beta}) - (Y_{p\beta} - Y_{q\beta})) / (S_{q\beta-1} - S_{\beta}).$$

end

if ($t_{xr} > 0$)

then if ($0 \leq t_{xc} \leq t_{xr}$) **then** { t_{xc} } \leftarrow { t_{xc} } + t_{xc} .

else if ($t_{xr} \leq t_{xc} \leq 0$) **then** { t_{xc} } \leftarrow { t_{xc} } + t_{xc} .

end

The new reference edge and reference vertex are determined by calling the Procedure Find_e_c-v_c provided in Section 4.2. The parameters depend on the situations. As illustrated in Figure 5.2, if the current reference edge belongs to P (case (a) and (b)), the parameters are (e_c, v_c, v_a, v_c). If the current reference edge belongs to Q (case (c) and (d)), the parameters are (e_c, v_c, v_c, v_β). As a result of calling the procedure, we get the new current reference edge and vertex.

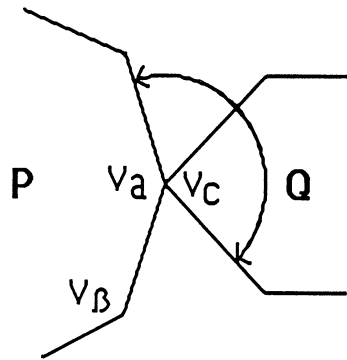
Once the new reference edge and vertex are determined, we calculate the X component of translated distance of Q along the new reference edge to the next refraction point, which is denoted by t_{xr}. Since the vertices of the current reference edge are v_a and v_β, t_{xr} is X_a - X_β.

<Insert Figure 5.2>

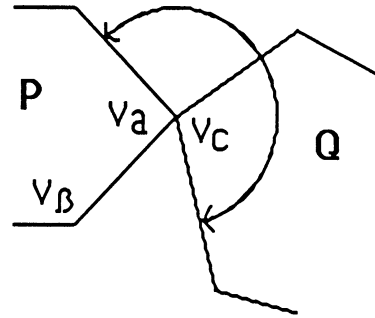
5.3 Updating the Slope of the Added Area Function

The next refraction point is the adjacent refraction point in the direction Q translates from the current refraction point. The next refraction point is the minimum (or maximum) of the elements of {t_{xc}} and t_{xr} if t_{xr} is positive (or negative). Corresponding to the refraction point, we update the slope of the added area function.

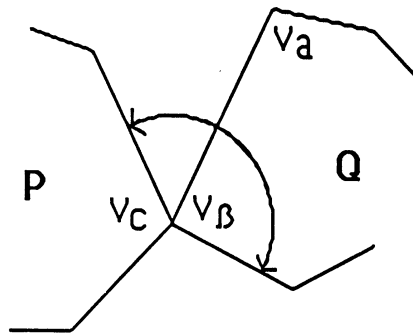
The updating of the slope of the added area function depends on the cause of the refraction point. If the next refraction point corresponds to a convex hull vertex change, we need to update the values of C_b and C_r. For the general case of this change, Procedure Update_Slope_CHV is



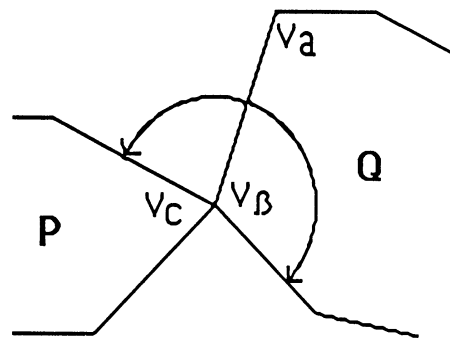
(a) $P \rightarrow P$



(b) $P \rightarrow Q$



(c) $Q \rightarrow P$



(d) $Q \rightarrow Q$

Figure 5.2 Finding New Reference Edge and Vertex

provided. In this procedure, t_{xc} stands for the next refraction point among the elements of the set $\{t_{xc}\}$. If the next refraction point corresponds to the reference edge change, we only need to change S_β . Even if the reference edge changes, the value of C_b or C_r does not change since it depends on only the convex hull vertices.

<Insert Procedure Update_Slope_CHV>

Note that, whenever Q is translated to the next refraction point, the origin of the added area function is moved along X-axis by X component of the translated distance. This enables us to use the current added area as a constant term in the added area function. Therefore, we only need to update the slope of the added area function. For the coordinates of vertices of P and Q , we keep the coordinates of the initial configuration. By keeping track the movement of origin and the translation of Q , we can calculate the coordinates of vertices we need in constant time. In the algorithm given in the next section, we assume the coordinates of P and Q is the updated coordinates.

5.4 The Algorithm and Its Time Complexity

The algorithm for finding the minimum area convex hull of two convex polygons is given as Algorithm 1. Using the procedures in Section 4, we initialize the configuration of P and Q and compute the initial added area. Then, we trace the refraction points to find the minimal added area and its configuration. In this Algorithm, Min (or Max) $\{t_{xc}\}$ finds the smallest (or largest) element in the set $\{t_{xc}\}$. A_{min} , me_c , mv_c , and mt_c are the value of A_c , e_c , v_c , and t_c , respectively, when the added area is minimal. Here, t_c is a variable to

Procedure Update_slope_CHV({CHV}, at_{xc}, C_b, C_r)

begin

case at_{xc} **of**

$$v_{pa} : C_b \leftarrow C_b + (X_{pa+1} - X_{pa})S_{pa}.$$

$$C_r \leftarrow C_r - (X_{pa+1} - X_{pa}).$$

$$v_{pa} \leftarrow v_{pa+1}.$$

$$v_{p\beta} : C_b \leftarrow C_b - (X_{p\beta+1} - X_{p\beta})S_{p\beta}.$$

$$C_r \leftarrow C_r + (X_{p\beta+1} - X_{p\beta}).$$

$$v_{p\beta} \leftarrow v_{p\beta+1}.$$

$$v_{qa} : C_b \leftarrow C_b - (X_{qa} - X_{qa-1})S_{qa-1}.$$

$$C_r \leftarrow C_r + (X_{qa} - X_{qa-1}).$$

$$v_{qa} \leftarrow v_{qa-1}.$$

$$v_{q\beta} : C_b \leftarrow C_b + (X_{q\beta} - X_{q\beta-1})S_{q\beta-1}.$$

$$C_r \leftarrow C_r - (X_{q\beta} - X_{q\beta-1}).$$

$$v_{q\beta} \leftarrow v_{q\beta-1}.$$

end

end

keep track of the X component of translated distance along e_c from the vertex v_β to the current refraction point.

Since the time to update the values of C_b , C_r , and S_β at each refraction point is constant, the time complexity of Algorithm 1 depends on the number of refraction points. The following Lemma 5.1 gives the limit of the number of refraction points.

Lemma 5.1 The maximum number of refraction points is $3N$.

[proof] From Lemma 3.2 and Lemma 3.3, we know that a refraction point is caused by a change in the convex hull vertex or a change in the reference edge. There are four convex hull vertices, two from P and two from Q. As Q translates along the boundary of P, every vertex of P and Q becomes a convex hull vertex exactly twice, once in the front added polygon and once in the rear added polygon. For the special case as illustrated in Figure 3.3, there is a reduction in the number of refraction points by at most 3. Hence, there can be no more than $2N$ refraction points due to a change in the convex hull vertex. If there are no special cases, the slope of the added area function changes whenever the reference edge changes. Since every edge of P and Q becomes a reference edge exactly once, the maximum number of refraction points due to a change in the reference edge is N . Consequently, the maximum number of refraction points is $3N$. ■

In Algorithm 1, we update the slope of the added area function whenever a convex hull vertex or the reference edge changes. Therefore, we update the slope exactly $3N$ times even if some of the updating is of

no use. We can state the following theorem which gives the time complexity of Algorithm 1.

Theorem 5.1 The time complexity of Algorithm 1 is linear in N .

Algorithm 1. Find_minimum_area_convex_hull

1. Initialization

Find the initial configuration of P and Q, and v_{pr} and v_{qr} .

Find $_{e_c-v_c}(e_i, v_c, v_{pr}, v_{qr})$.

$e_c \leftarrow e_i$.

Find $_{CHV-A_c}(\{CHV\}, A_c, v_{pr}, v_{qr})$.

$A_{min} \leftarrow A_c$, $me_c \leftarrow e_c$, $mv_c \leftarrow v_c$, $mt_c \leftarrow 0$.

Slope_of_added_area_function($\{CHV\}, C_b, C_r$).

2. Repeat

$t_c \leftarrow 0$.

$t_{xr} \leftarrow x_a - x_\beta$.

$\{t_{xc}\} \leftarrow \emptyset$.

for each CHV in $\{CHV\}$

 Calculate $_{t_{xc}}(CHV, \{CHV\}, S_\beta, t_{xr}, \{t_{xc}\})$.

while $\{t_{xc}\} \neq \emptyset$ do

 begin

 if ($t_{xr} > 0$)

 then $at_{xc} \leftarrow \text{Min}\{t_{xc}\}$.

 else $at_{xc} \leftarrow \text{Max}\{t_{xc}\}$.

$A_c \leftarrow A_c + (C_b + C_r S_\beta)at_{xc}/2$.

$t_c \leftarrow t_c + at_{xc}$.

 if ($A_c < A_{min}$) then

$A_{\min} \leftarrow A_c, me_c \leftarrow e_c, mv_c \leftarrow v_c, mt_c \leftarrow t_c.$

Update_slope_CHV({CHV}, at_{xc}, C_b, C_r).

$t_{xr} \leftarrow t_{xr} - at_{xc}.$

for each CHV in {CHV}

Calculate t_{xc} (CHV, {CHV}, $S_\beta, t_{xr}, \{t_{xc}\}$).

end

$A_c \leftarrow A_c + (C_b + C_r S_\beta) t_{xr} / 2.$

$t_c \leftarrow t_c + t_{xr}.$

if ($A_c < A_{\min}$) then

$A_{\min} \leftarrow A_c, me_c \leftarrow e_c, mv_c \leftarrow v_c, mt_c \leftarrow t_c.$

if ($e_c < P$)

then Find $e_{c-v_c}(e_c, v_c, v_a, v_c).$

else Find $e_{c-v_c}(e_c, v_c, v_c, v_\beta).$

Stop when $e_c = e_i.$

6. Conclusion

We have shown a linear time algorithm for finding the minimal area convex hull for two non-overlapping convex polygons P and Q under translation. This is done by tracing the refraction points of the added area function since the function is piecewise linear and the number of refraction points is linear in N .

If both rotation and translation are allowed, then the problem of finding the minimal area convex hull becomes more difficult. The added area function will be a three-dimensional surface with axis for area, translation and rotation. Because of rotation, the added area function is expected to be sinusoidal when projected. Refraction points becomes refraction curves. If there is linear number of such refraction curves then an algorithm with a lower bound of $O(N^2)$ time is conceivable since a linear number of refraction points along the translation axis is expected to remain. If the given polygons are not convex, the problem becomes even more difficult.

References

- [1] A. Albano and G. Sapuppo, "Optimal Allocation of Two-Dimensional Irregular Shapes Using Heuristic Search Methods," IEEE Trans. System, Man, and Cybernetics, Vol SMC-10, No. 5, pp. 242-248, May 1980.
- [2] B. Chazelle, "The Bottom-Left Bin-Packing Heuristic: An Efficient Implementation," IEEE Trans. Computers, Vol. C-32, No. 8, pp. 697-707, August 1983.
- [3] H. Freeman and R. Shapira, "Determining the Minimum Area Incasing Rectangle for an Arbitrary Closed Curve," Comm. ACM, Vol. 18, No. 7, pp. 409-413, July 1975.
- [4] P. C. Gilmore and R. E. Gomory, "Multistage Cutting Stock Problems of Two or More Dimensions," Operations Research, Vol. 13, pp. 94-120, 1965.
- [5] R. Graham and F. Yao, "Finding the Convex Hull of a Simple Polygon," J. of Algorithms, Vol. 4, pp. 324-331, 1983.
- [6] D. S. Johnson, "Near-Optimal Bin Packing Algorithms," Technical Report MAC TR-109, Project MAC, MIT, Cambridge, Massachusetts, 1973.
- [7] S. Israni and J. Sanders, "Two-Dimensional Cutting Stock Problem Research: Review and a New Rectangular Layout Algorithm," J. of Manufacturing Systems, Vol. 1, No. 2, pp. 169-182, 1982.
- [8] M. I. Shamos, "Geometric Complexity," Seventh Annual ACM SIGACT Conference, pp. 224-233, May 1975.

Appendix. Proof of Lemma 3.1

We compute the area of a translated (new) triangle as a function of the area of the old triangle. Here we use the same notations defined in Section 3.1. If the base edge of the old triangle belongs to P , the area of the new triangle $v_i v_j v_k'$ is:

$$\begin{aligned} & (v_i v_j \times v_j v_k')/2 \\ = & (((X_j, Y_j) - (X_i, Y_i)) \times ((X_k', Y_k') - (X_j, Y_j)))/2 \end{aligned}$$

Expressing X_k' as $(X_k + t_x)$ and Y_k' as $(Y_k + S_\beta t_x)$, we have:

$$= (((X_j, Y_j) - (X_i, Y_i)) \times ((X_k + t_x, Y_k + S_\beta t_x) - (X_j, Y_j)))/2$$

Expanding the cross product, we have:

$$\begin{aligned} = & ((Y_j - Y_i) (X_k - X_j) - (X_j - X_i) (Y_k - Y_j) + \\ & (Y_j - Y_i) t_x - (X_j - X_i) S_\beta t_x)/2 \end{aligned}$$

The sum can be expressed in terms of the area of the old triangle $v_i v_j v_k$ as:

$$\begin{aligned} = & ((X_j - X_i, Y_j - Y_i) \times (X_k - X_j, Y_k - Y_j))/2 + \\ & ((Y_j - Y_i) t_x - (X_j - X_i) S_\beta t_x)/2 \\ = & (v_i v_j \times v_j v_k)/2 + ((X_j - X_i) S_i t_x - (X_j - X_i) S_\beta t_x)/2 \\ = & (v_i v_j \times v_j v_k)/2 + (X_j - X_i) (S_i - S_\beta) t_x/2 \\ = & \text{area of the old triangle} + \text{area change} \end{aligned}$$

Similarly if the base edge of the old triangle belongs to Q , the area of the new triangle $v_i' v_j' v_k'$ can be expressed in terms of the area of $v_i v_j v_k$ as:

$$\begin{aligned} & (v_j' v_i' \times v_i' v_k')/2 \\ = & (v_j v_i \times v_i v_k)/2 + (x_j - x_i) (S_i - S_\beta) t_x/2 \end{aligned}$$

Since there are only two kinds of triangles in added area polygons, above results prove the Lemma 3.1. ■