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

Technical Report 86-35

September 1986

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

September 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 positions 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 determine the minimal area convex hull in O(N) time. Instead of computing the convex hull after every translation, we compute the slope of a function for the added area 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.

Key Words

Convex hull, Computational geometry, Minimum area, Algorithms.

AMS Subject Classifications

52A15 Convex sets in 2 dimensions

52A45 Packing, covering, tiling (05B40)

List of Symbols

AAF : Added area function.

A_c : Added area of the current configuration.

C_b : Sum of constants corresponding to the base edges in the slope formula of AAF.

CHV : Convex hull vertex.

{CHV} : Set of CHV.

C_r : Sum of constants corresponding to the current contact edge in the slope formula of AAF.

e_c : Current contact edge.

e_i : Initial contact edge.

mA_c : Minimal added area.

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.

nt_{xc} : Next refraction point in $\{t_{xc}\}$.

P, Q : Convex polygons.

S_i : Slope of the edge $v_i v_{i+1}$.

S_β : Slope of the contact edge e_c $(v_\alpha v_\beta)$.

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_α.

v_i : $i^{\underline{th}}$ vertex of P and Q for i=1,2,....,N.

$v_{p\alpha}(v_{q\alpha})$: 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).

$v_\alpha(v_\beta)$: Vertex of the contact edge e_c in the front (rear) added polygon.

v_c : Current contact vertex.

$X_i(Y_i)$: X or Y coordinates of v_i.

1. Introduction

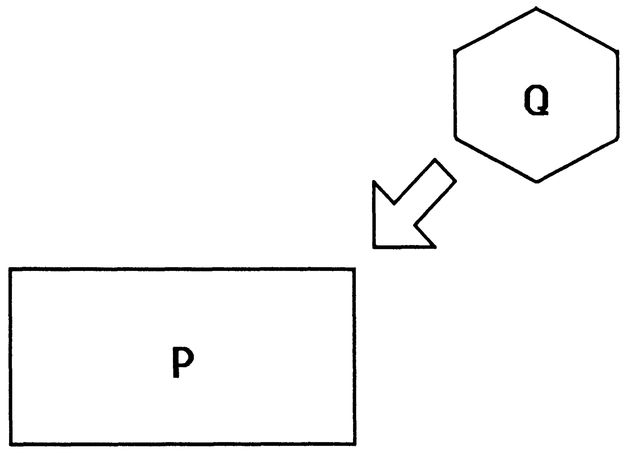
In computer science and operations research, allocating finite resources has been an important problem. This problem has many applications in industry and is characterized by the clustering of demands to minimize waste (or to maximize utility). The geometric problem of packing rectangular shapes is formulated as the Bin-Packing problem [6] and the Cutting-Stock problem [4]. It is NP-hard to determine optimal packing for rectangular shapes for which recent solutions have employed heuristics [2,7]. For non-rectangular shapes, it is reasonable to expect even greater difficulties. One approach to simplify the packing of non-rectangular shapes is to preprocess them by circumscribing with rectangles [1] and to pack the resulting rectangles [3]. Since rectangular circumscription produces more waste than convex circumscription, we develop a more space-efficient preprocessing algorithm by clustering the original shapes by two - specifically, two convex polygons into one with minimal convex hull. If a polygon is non-convex, we can apply a linear time algorithm to find its convex hull [5].

Given two convex polygons P and Q, we assume that Q is allowed to translate along the boundary of P as shown by the sequence in Figure 1. Suppose the minimum area convex hull occurs at a configuration in which a vertex from P touches a vertex from Q. There are O(N) such possible configurations, where N is the total number of vertices of P and Q. For each such configuration, their convex hull can be found in O(N) time. Hence, brute force leads to an algorithm that would run in at least $O(N^2)$ time. But, such an algorithm is not guaranteed to work. As shown in Figure 1(c), the minimum area convex hull occurs in a configuration other than the discrete vertex-vertex configuration.

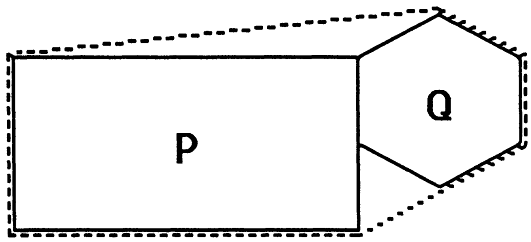
<Insert Figure 1>

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. Exploiting its piecewise linearity, we find the minimal added area configuration by calculating 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 the 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 the total number of refraction points to be O(N).

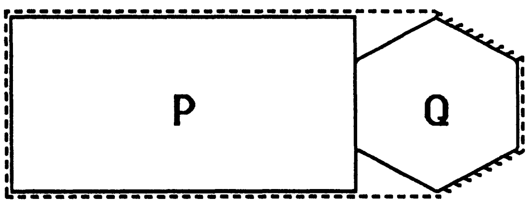
This paper is organized as follows. In section 2, definitions and notations are given. In section 3, the added area function is examined. In section 4, initialization processes are discussed. In section 5, the algorithm and its analysis are presented. In section 6, we conclude this paper.



(a) Given Two Polygons P and Q



(b) Vertex-Vertex Configuration is not necessarily Minimal Area Convex Hull



(c) Minimal Area Convex Hull

Figure 1. Brute Force Algorithm may not Lead to Optimal Solution

2. Preliminaries

In this section, we define terms related to the convex hull of polygons P and Q, with Q being allowed to translate counter-clockwise along the boundary of P. Among a total of N vertices, n of them belong to P. We indicate the $i^{\underline{th}}$ vertex of P, starting from a fixed vertex and indexing in a counter-clockwise order, by v_i where i=1,2,....,n. The vertices v_i of Q are indexed in the clockwise order, where i=n+1,n+2,....,N. (The reason for using the opposite ordering is to simplify the equation of the added area functions later.) The vertices of P and Q are stored in two doubly linked lists. We indicate the slope of the edge $v_i v_{i+1}$ by S_i.

Consider the convex hull over two convex polygons P and Q in contact as illustrated in Figure 2. There exist two added polygons which constitute the difference between the convex hull and the polygons P and Q. Based on the direction of travel, one of the added polygons is in the front and the other one is in the rear. The total area of these two added polygons is the added area, which can be zero or positive. The two polygons touch each other, with one providing an edge and the other a vertex. The edge, along which Q translates, is called the current contact edge e_c with slope S_β. The vertex v_c from the other polygon is called the contact vertex. In Figure 2, a contact edge is marked by a double line and a contact vertex by a circle.

<Insert Figure 2>

To form a convex hull of two convex polygons, new convex hull edges are needed. The new edge in the front connects two convex hull

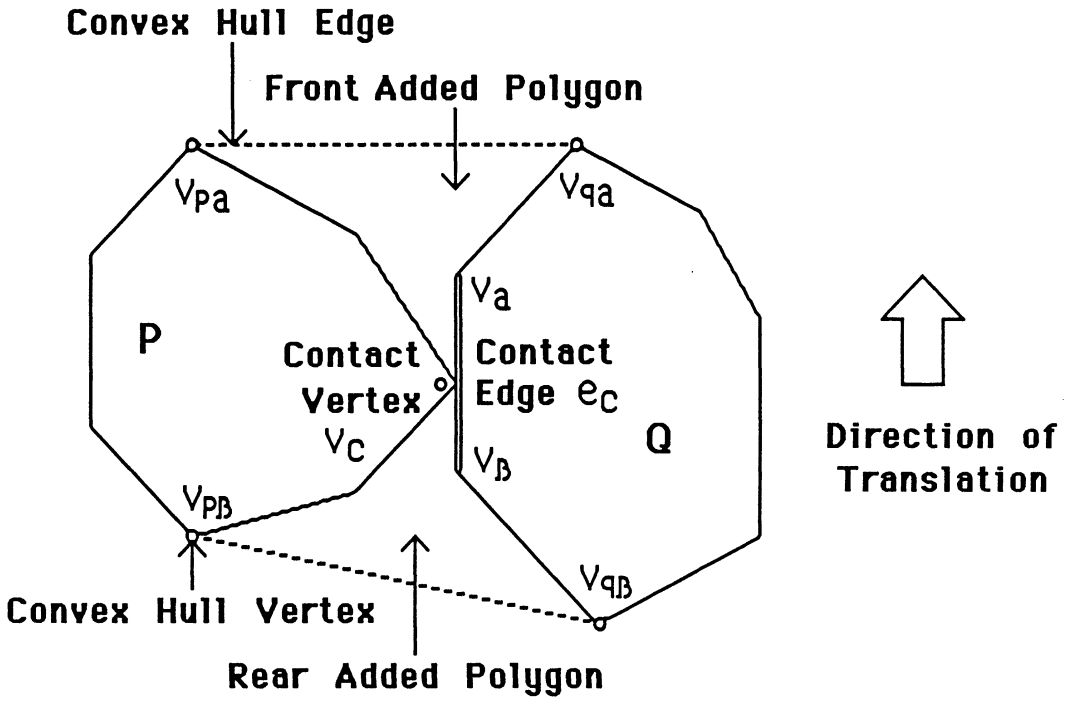


Figure 2. Convex Hull of Two Polygons

vertices, v_{pa} and v_{qa}. Similarly, there is one in the rear. If the convex hull vertices from P and Q coincide, we count them as two distinct vertices. Therefore, we can assume that for the general case there are always four convex hull vertices joined by two convex hull edges.

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 contact 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 by triangle area functions, the sum of which is the added area function.

In general, a triangle in a triangulated added polygon shares an edge with polygon P or Q. The shared edge will be referred to as a base edge. In some cases, triangles share two edges, one with P and the other with Q, which are shaded in Figure 3. Selecting the base edge then depends on which polygon provides a contact edge. If the contact 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 contact edge belongs to Q, we select the base edge in the opposite way. A base edge is marked by a double line in Figure 3.

<Insert Figure 3>

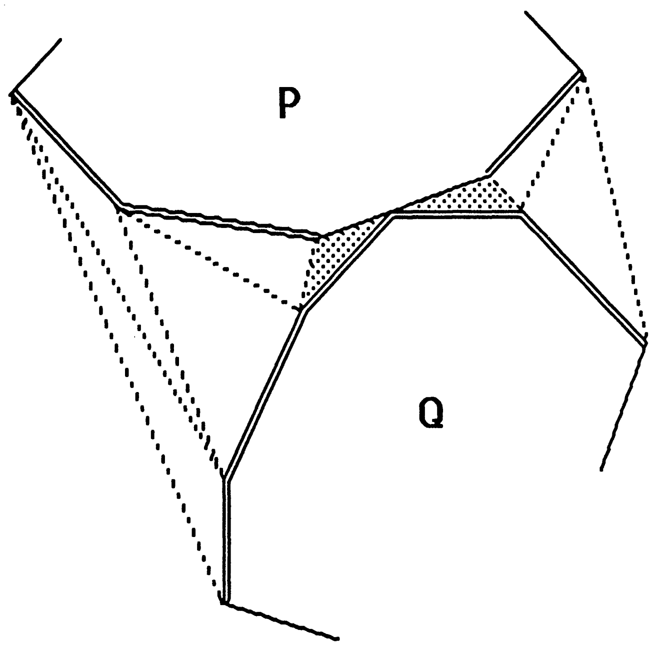


Figure 3. Base Edges (double lined) of
Triangles in the Added Polygons

3. Added Area Function

In this section, we examine the added area function (AAF). The added area is the sum of triangles since it can be triangulated. To characterize AAF, first we prove that the triangle area function is linear. Based on its linearity, we derive the slope of AAF. (The reason we are interested only in the slope is due to the following observation. Whenever Q is translated by t_x, the origin of AAF can be translated by the same amount. This enables us to treat the current added area as a constant term in AAF.) By examining a formula for AAF, we state the conditions when its slope changes. Finally, the piecewise linearity of AAF is asserted.

As Q translates along the boundary of P, a triangle in the added area polygon exists for a certain range of the translated distance. Since the added area is the sum of the triangles in the added polygons, we can calculate AAF from the triangle area functions.

Consider the area of a triangle $v_i v_j v_k$ when polygon Q is translated by (t_x, t_y) along the current contact edge e_c $(v_\alpha v_\beta)$ with slope S_β. The area of the triangle $v_i v_j v_k$ is computed by the cross product of its two edges using the left hand rule. The cross product $(v_i v_j \times v_j v_k)/2$ is the area of the 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 Figure 4, the base edge $v_i v_j$ with slope S_i belongs to Q and v_k belongs to P.

<Insert Figure 4>

We now establish a linear relationship between area and the amount

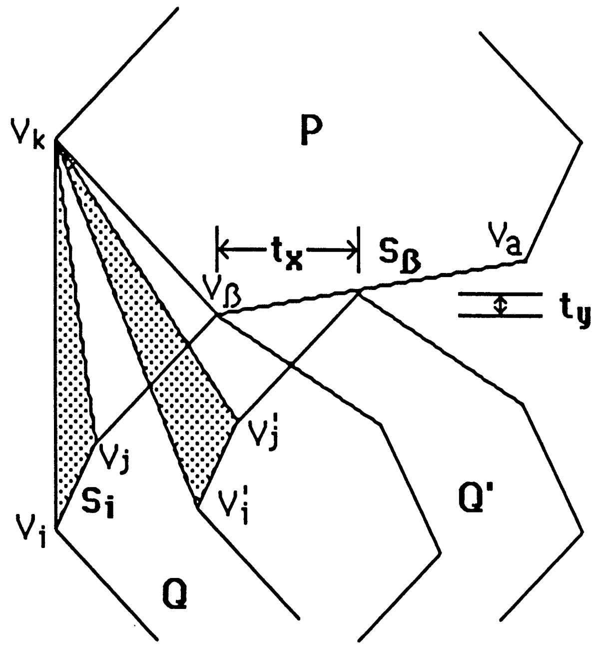


Figure 4. Calculation of the Area of a Triangle

of translation by assuming that no edge has an infinite slope. (The detail for resolving an infinite slope is discussed in Section 4.)

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

$$(X_j - X_i)(S_i - S_\beta)/2, \tag{1}$$

where S_β is the slope of e_c and S_i is the slope of edge $v_i v_j$ with X-coordinates X_i and X_j.

[Proof] The proof is given in Appendix 1.

The slope of AAF depends on the slope of the base edges and the slope of the contact edge since the slope of a triangle area function depends on the same quantities from Lemma 3.1. It is worthwhile to note that the slope of AAF does not depend on how an added polygon is triangulated.

The calculation for the slope of AAF follows. We know that the sequence of edges between $v_{p\beta}$ and $v_{p\alpha}$ and those between $v_{q\beta}$ and $v_{q\alpha}$ are the base edges, where $v_{p\alpha}$ (or $v_{q\alpha}$) 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 the current contact edge is e_c $(v_\alpha v_\beta)$, the slope of AAF is:

$$\sum_{i=p\beta}^{p\alpha-1} (X_{i+1} - X_i)(S_i - S_\beta)/2 + \sum_{i=q\beta}^{q\alpha-1} (X_{i+1} - X_i)(S_i - S_\beta)/2 \tag{2}$$

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

Let us turn to the changes in the slope of AAF. From formula (2), we see that the slope changes if S_β changes or if the number of terms in the summation changes. The latter case corresponds to the situation when the convex hull vertex changes and the former case when the contact edge changes. The following two lemmas clarify these situations.

Lemma 3.2 The slope of AAF changes if the coefficient of S_β is non-zero and the slope of the contact edge changes.

[proof] In formula (2) for the slope of AAF, the coefficient of S_β is $(X_{p\beta} - X_{p\alpha}) + (X_{q\beta} - X_{q\alpha})$. This coefficient is non-zero by assumption. Therefore, if the value of S_β, the slope of the contact edge, changes, the slope of AAF also changes. ∎

Lemma 3.3 The slope of AAF changes if a convex hull vertex changes.

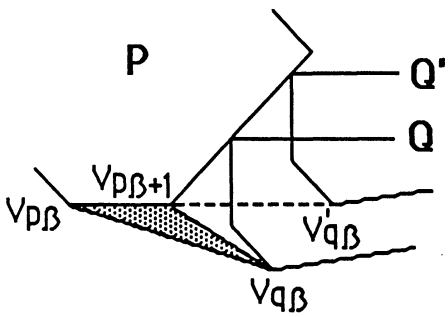
[Proof] If a convex hull vertex changes, the number of terms in formula (2) increases or decreases by one. In either case, the slope of AAF changes because the slope of the triangle area function of the appearing (or disappearing) triangle cannot be zero. Because both P and Q are convex polygons, the slope of the base edge cannot be the same as the slope of the contact edge. Hence, by Lemma 3.1, the change in the slope of AAF due to an appearing (or a disappearing) triangle must be nonzero. ∎

Changes in the convex hull vertex are illustrated in Figure 5. In Figure 5(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 $v_{p\beta}v_{p\beta+1}$. Similarly, in Figure 5(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. It is possible that more than one convex hull vertex changes simultaneously as illustrated in Figure 5(c). The slope of AAF, then, may not change at all since it is possible that the sum of the added terms equals the sum of the deleted terms. To resolve such a case, we consider the changes of convex hull vertices one at a time.

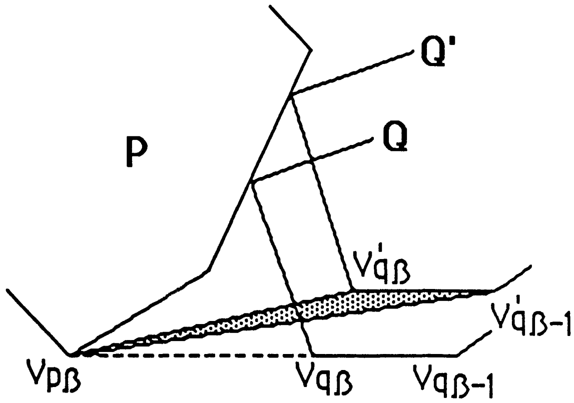
<Insert Figure 5>

Now we are ready to summarize the shape of AAF. Lemma 3.2 and Lemma 3.3 imply that AAF is piecewise linear as there exist points where the slope of AAF changes. There cannot be a jump in AAF where the slope changes. If that were the case, the area at that point on the AAF would be multi-valued. This implies that the convex hull over P and Q for a particular configuration is non-unique which leads to a contradiction. Therefore, AAF is continuous at points where the slope changes. Hence, AAF is continuous piecewise linear.

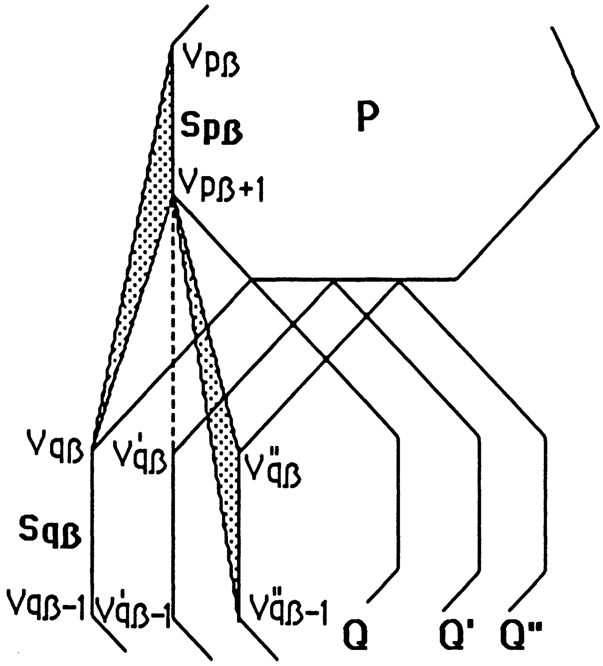
Since the area of the polygons P and Q is constant, finding the minimal area convex hull of P and Q is equivalent to finding the minimum in AAF. Since AAF is piecewise linear the minimum of AAF is at one of the refraction points where the slope of AAF changes. In Section 5, we develop an algorithm to find the minimal convex hull of P and Q by tracing AAF at these points. It is preceded by some initialization procedures in Section 4.



(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.



(c) 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 5. Changes of Convex Hull Vertices

4. Initialization

Before finding the minimal area convex hull, we determine the initial values of the parameters to be used in the algorithm in Section 5. First, we set the initial configuration of P and Q. Secondly, a procedure to determine the initial contact edge and contact vertex is given. Thirdly, we calculate the initial added area and determine the initial convex hull vertices. And finally, we calculate the initial slope of AAF. All of the above are done in linear time and the procedures are given in Appendix 2.

4.1 Initial Configuration of P and Q

We find the initial configuration by translating Q so that a vertex of P and a vertex of Q coincide. 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 vertex, choose the one with the largest X-coordinate as v_{pr} or the one with the smallest X-coordinate as v_{qr}.) We translate Q such that v_{qr} coincides with v_{pr}. We assume there is no edge with an infinite slope. If there is such an edge, rotating the polygons by a "small" amount about v_{pr} eliminates the infinite slope. These steps are performed in linear time.

4.2 Initial Contact Vertex and Contact Edge

The initial contact edge e_i and the initial contact vertex v_c may belong to P or Q. To find them we use the Procedure Find_e_c_v_c listed in Appendix 2. The input of the procedure is v_{pc} and v_{qc}, the two contact vertices from P and Q. They are initially set at v_{pr} and v_{qr}, respectively.

The procedure works as follows. If the left-handed cross product $v_{qc-1}v_{qc}$ x $v_{pc}v_{pc+1}$ is positive, the counter-clockwise angle of $v_{qc-1}v_{qc}v_{pc+1}$ is less than π. In this case, the current contact edge e_c is $v_{pc+1}v_{pc}$ which belongs to P and the current contact vertex v_c is v_{qc} which belongs to Q. If the cross product is not positive, e_c is $v_{qc}v_{qc-1}$ and v_c is v_{pc}. The same procedure is also used for finding the next contact vertex and contact edge.

4.3 Initial Convex Hull Vertices and Added Area

While we find the initial convex hull vertices, we also calculate the initial added area in Procedure Find_CHV_A_c. The output of the procedure are {CHV}, a set of four convex hull vertices, $v_{p\alpha}$, $v_{p\beta}$, $v_{q\alpha}$, and $v_{q\beta}$, and the current added area A_c. The input of the procedure are the two contact vertices v_{pr} and v_{qr}.

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

In performing the Procedure Find_CHV_A_c, each 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 formula (2) in Section 3. To simplify updating, we use the following representation for the slope of AAF:

$$(C_b + C_r\ S_\beta)/2$$

where C_b = Sum of constants corresponding to the base edges

C_r = Sum of constants corresponding to the contact edge

S_β = Slope of the contact edge.

Then the slope of AAF has terms:

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

$$C_r = -\sum_{i=p\beta}^{p\alpha-1} (X_{i+1} - X_i) - \sum_{i=q\alpha}^{q\beta-1} (X_{i+1} - X_i)$$

$$= (X_{p\beta} - X_{p\alpha}) + (X_{q\beta} - X_{q\alpha}).$$

5. The Algorithm and Its Analysis

We are ready to present an algorithm that gives the convex hull for the two convex polygons P and Q for which the added area is minimal. The AAF is continuous piecewise linear and its slope changes at refraction points. To find the minimal area configuration, we trace only the refraction points by updating the slope in constant time. Then, the algorithm would run in linear time if there is a 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. We begin with the updating for the refraction points needed by the algorithm.

5.1 Refraction Point Due to Contact Edge Change

From Lemma 3.2, when the contact edge changes, there exists a refraction point. Correspondingly, we determine the new contact edge and contact vertex. Then, we calculate the maximum distance Q can be translated along the new contact edge to reach the next refraction point.

The new contact edge and contact vertex are determined by calling the Procedure Find_e_c_v_c explained in Section 4.2. The parameters depend on the situation. If the current contact edge belongs to P, the parameters are (e_c, v_c, v_α, v_c). If the current contact edge belongs to Q, the parameters are (e_c, v_c, v_c, v_β).

Once the new contact edge and vertex are determined, we calculate the X component of translated distance of Q along the new contact edge to the next refraction point, which is denoted by t_{xr}. Since the vertices of the current contact edge are v_α and v_β, t_{xr} is $X_\alpha - X_\beta$.

5.2 Refraction Point Due to Convex Hull Vertex Change

When a convex hull vertex changes, the slope of AAF changes as well. We calculate the X component of the translated distance denoted by t_{xc} along the current contact edge to reach the next refraction point.

<Insert Figure 6>

In Figure 6, 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 the edge $v'_{q\beta-1}v'_{q\beta}$ is:

$$Y - Y'_{q\beta} = S_{q\beta-1}(X - X'_{q\beta}). \tag{3}$$

By replacing $X'_{q\beta}$ with $(X_{q\beta} + t_{xc})$ and $Y'_{q\beta}$ with $(Y_{q\beta} + S_\beta t_{xc})$ and by inserting the coordinates of $v_{p\beta}$ into equation (3), we get the refraction point due to the change of $v_{q\beta}$:

$$t_{xc} = (S_{q\beta-1}(X_{p\beta} - X_{q\beta}) - (Y_{p\beta} - Y_{q\beta}))/(S_{q\beta-1} - S_\beta). \tag{4}$$

For the general case, we use Procedure Calculate_t_{xc} in Appendix 2. The term t_{xr} in the procedure is explained in the previous section.

5.3 Updating the Slope of the Added Area Function

The next refraction point nt_{xc} is the minimum or maximum of the

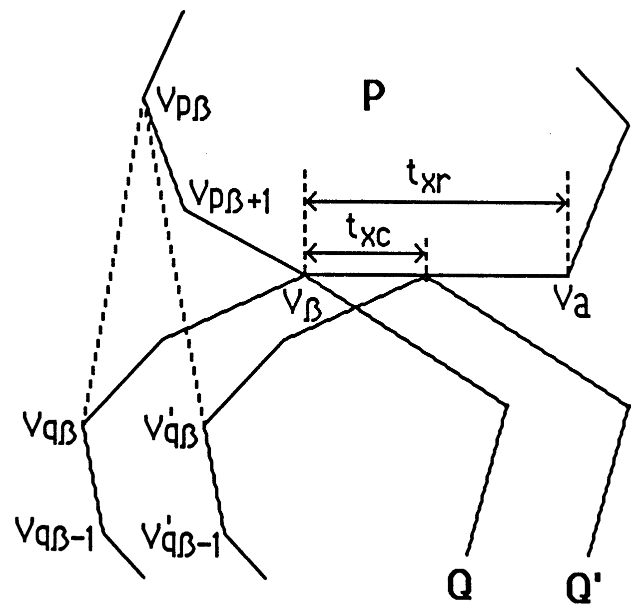


Figure 6. Refraction Point Due to a Change in the Convex Hull Vertex

elements of $\{t_{xc}\}$ and t_{xr} depending on whether t_{xr} is positive or negative. Corresponding to that refraction point, we update the slope of AAF using Procedure Update_slope_CHV in Appendix 2. The updating depends on the cause of the next refraction point. If it is caused by a change in a convex hull vertex, we update the values of C_b and C_r. If it is caused by a change in the contact edge, we only need to change S_β, since C_b and C_r depend on only the convex hull vertices.

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 $\{t_{xc}\}$ finds the smallest element in the set $\{t_{xc}\}$; similarly, for Max $\{t_{xc}\}$. The values mA_c, me_c, mv_c, and mt_c store the values of A_c, e_c, v_c, and t_c, respectively, when the added area is minimal. Additionally, t_c is a variable for the X component of translated distance along e_c from the vertex v_β to the current refraction point. For the vertices of P and Q, we keep their coordinates from the initial configuration and update them. By keeping track of the amount of translation for Q, hence the translation of the coordinate system, we calculate the coordinates of a vertex under consideration in constant time.

<Insert Algorithm 1>

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

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}).

$mA_c \leftarrow A_c$, $me_c \leftarrow e_c$, $mv_c \leftarrow v_c$, $mt_c \leftarrow 0$.

Slope_of_added_area_function({CHV}, C_b, C_r).

2. Repeat

$t_c \leftarrow 0$.

$t_{xr} \leftarrow X_\alpha - X_\beta$.

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

for each CHV in {CHV}

Calculate_t_{xc}(CHV, {CHV}, S_β, t_{xr}, $\{t_{xc}\}$).

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

begin

if ($t_{xr} > 0$)

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

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

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

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

if ($A_c < mA_c$) **then**

$mA_c \leftarrow A_c$, $me_c \leftarrow e_c$, $mv_c \leftarrow v_c$, $mt_c \leftarrow t_c$.

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

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

for each CHV in {CHV}

Calculate_t_{xc}(CHV,{CHV},S_β,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 $(A_c < mA_c)$ **then**

$mA_c \leftarrow A_c$, $me_c \leftarrow e_c$, $mv_c \leftarrow v_c$, $mt_c \leftarrow t_c$.

if $(e_c \subset P)$

then Find_e_c_v_c(e_c,v_c,v_α,v_c).

else Find_e_c_v_c(e_c,v_c,v_c,v_β).

Stop when $e_c = e_i$.

on the number of refraction points. The following lemma gives the upper limit on that number.

<u>Lemma 5.1</u> 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 either a change in the contact edge or a change in the convex hull vertex. Since every edge of P and Q becomes a contact edge exactly once, the maximum number of refraction points due to a change in the contact edge is N. 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. Though, for the special case illustrated in Figure 5(c), 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. Consequently, the maximum number of refraction points is 3N. ∎

In Algorithm 1, we update the slope of AAF whenever a convex hull vertex or the contact edge changes. By Lemma 5.1, the slope is updated 3N times. We can now state the theorem which gives the time complexity of Algorithm 1.

<u>Theorem 5.1</u> The time complexity of Algorithm 1 is linear in N.

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 AAF 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 axes for area, translation and rotation. Because of rotation, AAF is expected to be sinusoidal when projected. Refraction points becomes refraction curves. If there is a 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, SMC-10 (1980), pp. 242-248.

[2] B. Chazelle, The Bottom-Left Bin-Packing Heuristic: An Efficient Implementation, IEEE Trans. Computers, C-32 (1983), pp. 697-707.

[3] H. Freeman and R. Shapira, Determining the Minimum Area Incasing Rectangle for an Arbitrary Closed Curve, Comm. ACM, 18 (1975), pp. 409-413.

[4] P. C. Gilmore and R. E. Gomory, Multistage Cutting Stock Problems of Two or More Dimensions, Operations Research, 13 (1965), pp. 94-120.

[5] R. Graham and F. Yao, Finding the Convex Hull of a Simple Polygon, J. of Algorithms, 4 (1983), pp. 324-331.

[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, 1 (1982), pp. 169-182.

Appendix 1. 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. If the base edge of the old triangle belongs to P, the area of the new triangle $v_i v_j v_k'$ is:

$$(v_i v_j \times v_j v_k')/2$$

$$= \quad (((X_j, Y_j) - (X_i, Y_i)) \times ((X_k', Y_k') - (X_j, Y_j)))/2$$

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

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

Expanding the cross product, we have:

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

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

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

= area of the old triangle + area change

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 $v_iv_jv_k$ as:

$$(v_j'v_i' \times v_i'v_k)/2$$

$$= \quad (v_jv_i \times v_iv_k)/2 + (X_j - X_i)\ (S_i - S_\beta)\ t_x/2$$

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

Appendix 2. Procedures

Procedure **Find_e_c_v_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

```
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 x 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 x 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 x v_qc v_pc.
            end
         p_cross_product ← v_qc v_pc x 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 }
   v_pc ← v_pr, v_qc ← v_qr.
   p_cross_product ← v_pc-1 v_pc x v_qc v_qc-1.
   if (p_cross_product > 0) then v_pc ← v_pc-1.
   while (p_cross_product > 0) do
      begin
         q_cross_product ← v_pc v_qc x v_qc v_qc-1.
         while (q_cross_product > 0) do
            begin
               A_c ← A_c + q_cross_product/2, v_qc ← v_qc-1.
               q_cross_product ← v_pc v_qc x v_qc v_qc-1.
            end
         p_cross_product ← v_pc-1 v_pc x v_pc v_qc.
         if (p_cross_product > 0) then
            A_c ← A_c + p_cross_product/2, v_pc ← v_pc-1.
      end
   v_pβ ← v_pc, v_qβ ← v_qc.
   {CHV} ← {CHV} + {v_pβ, v_qβ}.
end
```

```
Procedure  Slope_of_added_area_function({CHV}, C_b, C_r)

begin

   C_r ← (X_pβ - X_pα) + (X_qβ - X_qα)

   C_b ← 0.

   i ← v_pβ.

   while (i < v_pα) do

      begin

         C_b ← C_b + (X_i+1 - X_i) S_i.

         i ← i + 1.

      end

   i ← v_qβ.

   while (i < v_qα) do

      begin

         C_b ← C_b + (X_i+1 - X_i) S_i.

         i ← i + 1.

      end

end
```

```
Procedure  Calculate_t_xc(CHV,{CHV},S_β,t_xr,{t_xc})

begin

   case CHV of

      v_pα : t_xc = (S_pα(X_qα - X_pα) - (Y_qα - Y_pα)) / (S_β - S_pα).

      v_pβ : t_xc = (S_pβ(X_qβ - X_pβ) - (Y_qβ - Y_pβ)) / (S_β - S_pβ).

      v_qα : t_xc = (S_qα-1(X_pα - X_qα) - (Y_pα - Y_qα)) / (S_qα-1 - S_β).

      v_qβ : t_xc = (S_qβ-1(X_pβ - X_qβ) - (Y_pβ - Y_qβ)) / (S_qβ-1 - S_β).

  end

  if (t_xr > 0)

     then if (0 ≤ t_xc ≤ t_xr) then {t_xc} ← {t_xc} + t_xc.

     else if (t_xr ≤ t_xc ≤ 0) then {t_xc} ← {t_xc} + t_xc.

end
```

Procedure **Update_slope_CHV**($\{CHV\}, nt_{xc}, C_b, C_r$)

begin

case nt_{xc} **of**

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

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

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

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

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

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

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

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

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

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

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

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

end

end