Overlapping Area Computation between Irregular Polygons for Its Evolutionary Layout Based on Convex Decomposition

Zi-qiang Li, Yan He
School of Information & Engineering of Xiangtan University Xiangtan, Hunan 411105, China
Email: xtulzq@hotmail.com

Zhuo-jun Tian
Key Laboratory of Intelligent Computing & Information Processing of Ministry of Education, Xiangtan, 411105, China
Email: tianzj47@sina.com

***Abstract*—Low efficiency of interference calculation has become the bottleneck that restricts further development of the performance of evolutionary algorithm for the polygon layout. To solve the problem, in this paper, we propose an algorithm of calculating overlapping area between two irregular polygons. For this algorithm, at first, two irregular polygons are respectively decomposed into the minimum number of convex polygons; afterwards, each pair of the overlapping convex polygons from two resulting partitions is clipped and their overlapping area is calculated. Because through a fast non-overlapping test to all pairs of convex polygons to be clipped, invalid computation is decreased; by making use of simple internal vertex judgment and shear-transformation based on intersecting test and intersection calculation between a line segment and any convex broken line segment, its speed of clipping overlapping convex polygons is improved. The time complexity analysis and numerical experiments indicate that the performance of our presented algorithm superiors to the existing algorithms.**

***Index Terms*—Overlapping area calculation; Irregular polygon; Convex polygon Clipping; Evolutionary layout.**

I. INTRODUCTION

In the fields of computer graphics, computer-aided design, robotics and virtual reality etc., the non interference detection[1][2][3] and the overlapping area calculating [4][5] of the irregular polygons are often encountered. For example, at each iterative for evolutionary layout of irregular polygons, the sum of their overlapping area is computed and is used to evaluate fitness of population individual. Its computation time occupies very large proportion of entire optimization time for solving the layout problems. So, it has been being studied for many years. Then algorithms based on geometric element decomposition are presented. For example, Liu De-quan [4] and Wang Jin-min[5] publish algorithms for calculating the overlapping area of irregular polygons based on triangle and trapezium decomposition respectively. The directly clipping algorithms [6][7][8] also can be applied to compute the overlapping area of the irregular polygons. However, for layout schemes of individuals, the overlapping objects become fewer and fewer and the sum of their overlapping area gets smaller and smaller in the process of evolutionary iteration. Directly using them will go against improvement of interference calculation efficiency. In addition, directly clipping algorithms all have high time complexity (the retailed analysis sees Section V). So, algorithms for determining overlapping boundary of two irregular polygons based on convex decomposition are studied currently. For example, Li Jing et al[9] propose a clipping algorithm of irregular polygons based on monotone piece partition but the number of monotone pieces it results is larger than that of [10] or [11] results in most of the cases. On basis of convex decomposition and clipping, this paper presents an algorithm for computing irregular polygon overlapping area (CIPOA algorithm). Its convex decomposing and polygon clipping strategy are different from the existing algorithms respectively. Numerical experiments show that our CIPOA algorithm improves the calculation efficiency.

This paper is organized as follows: in this section, we give a brief overview of our CIPOA algorithm. A method of decomposing irregular polygons is described in section II. A clipping method for convex polygons is proposed in section III. Section IV describes our CIPOA algorithm. Three test examples are given in Section V. Finally, Section VI is a summarization of the paper.

This work is supported by Hunan Provincial Natural Science Foundation of China (11JJ6050) and Research Foundation of Education Bureau of Hunan Province, China (11A120) and the Doctor Startup Foundation of Xiangtan University of China (Grant No.09QDZ18).

II. CONVEX DECOMPOSITION OF THE IRREGULAR POLYGON

For quickly calculating the overlapping area between two irregular polygons through convex partition, the efficient decomposing strategy is its premise. For the irregular polygon decomposition, two kinds of algorithms have been published. (i) Native partition algorithms based on geometric element. For example, Ref.[4][5] describe the triangle and trapezium partition algorithm respectively. They are easy to be implemented. But in general cases, because the number of triangles or trapeziums to a partition is large, the efficiency of computing overlapping area is decreased. (ii) Convex decomposition algorithms with Steiner points and that without Steiner points. For the former, in 1985, Chazelle [10] constructed X_k-pattern of

doi:10.4304/jsw.7.2.485-492

reflex vertex group to partition it into minimum convex parts in $O(n+N^3)$ time; In 1996, Xiao Zhong-hui et al[11] proposed coding partition algorithm with $O(n+N^2)$ time. For the latter, a dynamic programming algorithm of Keil[12][13] computes a minimum composition in $O(nN^2 \log N)$ time; a partition algorithm with $O(n+N^2 \min(N^2,n)$ time is described in [14]; Wang Zheng-Xuan [15] presents weighted partition algorithm; the monotone-partitioning strategy[16] is applied to the irregular polygon clipping in [9]. It is known from [10][11] that algorithms with Steiner points may result the minimum number of convex polygons. Therefore an irregular polygon convex decomposition method with Steiner point (IPSPCD method) is proposed in this Section. It can product the minimum number of convex polygons and improve composition efficiency of irregular polygons.

A. Related Knowledge

Definition 1 Let P be an irregular polygon with n vertices $p_1,p_2,\ldots,p_n$, in clockwise order and satisfy following properties: (1) All vertices are different from each other; (2) Each vertex only belongs to the edge where it lies; (3) Any two edges of P don't intersect. A convex decomposition of P is a set of convex polygons whose union is P.

Definition 2 Let P be an irregular polygon and α be an interior angle consisted of its vertex p and two edges adjacent with p. If α<180°, then p is called convex, otherwise, is called *reflex*.

Rule 1 Let P be a polygon with n vertices, $p_{k-1}(x_{k-1},y_{k-1})$, $p_k(x_k,y_k)$ and $p_{k+1}(x_{k+1}, y_{k+1})(1\leqslant k\leqslant n)$ be three adjacent vertices of P, we set $s_k=(x_{k-1}-x_{k+1})(y_k-y_{k+1})-(y_{k-1}-y_{k+1})(x_k-x_{k+1})$. According to Section 2 in [11], if $s_k>0$, then p_k is a convex vertex, otherwise, if $s_k<0$, then p_k is a *reflex* vertex.

Definition 3 Let P be an irregular polygon with N *reflex* vertices (N>0), $p_k(x_k,y_k)$ be a *reflex* vertex of P ($1\leqslant k\leqslant n$). (ⅰ) For $p(x,y)$ on p_ip_{i+1}(i=1,2,⋯,n and $i\neq k$-1,k), if by connecting p and p_k, P is divided into two convex polygons for N =1,2 or a convex polygon and a non-convex polygon whose number of *reflex* vertices is less than or equal to N-1 for N>1, then p_kp_i is called the subdivision line segment of P; (ⅱ) if $q(x',y')$ lie in the interior of P, the broken line segment consists of line segment p_kq and qp_{k+1}(p_{k+1} is its *reflex* vertex) divides it into a convex polygon and a non-convex polygon whose number of *reflex* vertices is less than or equal to N-1, then p_kqp_{k+1} is the broken subdivision segment of P.

From definition 3 we can derive following two decomposition properties for the irregular polygon.

Property 1. Let P be an irregular polygon, p_i and p_j be its two consecutive *reflex* vertices in clockwise order. (i) If an open line segment connecting p_i and p_j lies in the interior of P and 2 adjacent edges of p_i or p_j don't lie in the same side of line p_ip_j, then p_ip_j is its subdivision line segment; (ii) if the extension line of $p_{i-1}p_i$ or $p_{j+1}p_j$ intersects with the broken line segment $p_ip_{i+1}\cdots p_{j-1}p_j$ at q and open line segment p_iq or p_jq lie in interior of P, then p_ip_j is its subdivision line segment.

Property 2. Let both p_i and p_{i+1} be the *reflex* vertices of an irregular polygon P. If the extension lines of $p_{i-1}p_i$ and $p_{i+2}p_{i+1}$ intersect at q and $\triangle p_ip_{i+1}q_i \subset P$, then the broken line segment p_iqp_{i+1} is its subdivision segment.

B. Convex polygon decomposition

Thought of Method Let P be the irregular polygon with N *reflex* vertices, then the decomposition of P can deal with as following three cases.

Case 1: when N = 1, Let $p_k(1\leqslant k\leqslant n)$ be only *reflex* vertex of P. Starting from p_k in clockwise order around P, we use STIC method(see section III) to certainly can search out edge p_ip_{i+1} (if i=n, then p_{i+1}=p_1) which the line $p_{k-1}p_k$ intersect with at $q(x,y)$. Therefore, by connecting p_k with q, P can be decomposed into two convex polygons according to definition 3(i).

Case 2: when N >1, both p_i and $p_{i+1}(1\leqslant i\leqslant n)$ are *reflex* vertex, if using STIC method, we can confirm that line $p_{i-1}p_i$ intersects with line $p_{i+2}p_{i+1}$, compute the x coordinate and the y coordinate of intersection q and both line segment p_iq and $p_{i+1}q$ lie in the interior of P except endpoint p_i and p_{i+1}, then According to property 2, P can be decomposed.

Case 3: when N >1, if $p_i(x_i, y_i)$ $(1\leqslant i\leqslant n)$ be a *reflex* vertex and p_{i+1} is a convex point. Starting from p_i, then we can search the next *reflex* vertex p_j of p_i in clockwise order around P. In order to describe conveniently, we give 4 conditional propositions as follows:

Proposition 1. Open line segment p_ip_j lies in the interior of P;

Proposition 2. $f_{ij}(p_{i-1})f_{ij}(p_{j+1})\geqslant 0$ and $f_{ij}(p_{i+1})f_{ij}(p_{j-1})\geqslant 0$(See Fig. 1(a)).

Proposition 3. $f_{ij}(p_{j-1})f_{ij}(p_{j+1})\leqslant 0$ and $f_{ij}(p_{i-1})f_{ij}(p_{j+1})\leqslant 0$(or $f_{ij}(p_{i-1})f_{ij}(p_{i+1})\leqslant 0$ and $f_{ij}(p_{i-1})f_{ij}(p_{j+1})$ $\leqslant 0$)(See Fig. 1 (b) and (c));

Note: In order to describe conveniently, $f_{ij}(p_k)$ given by (1) is used in Proposition 2 and 3. But their test method is STIC in practice.

$$f_{ij}(p_k)=\begin{cases} x_k-x_i & x_k-x_i=0 \\ y_k-y_i-\dfrac{y_k-y_i}{x_k-x_i}(x_k-x_i) & x_k-x_i\neq 0 \end{cases} \qquad (1)$$

$$k=i-1,i+1,j-1,j+1$$

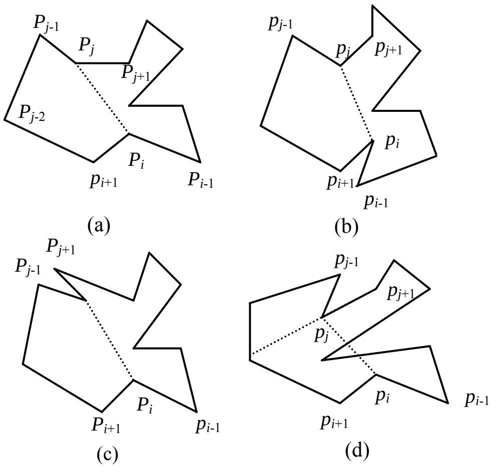


Fig.1 4 Cases of *reflex* vertex decomposition for polygon

Proposition 4. $p_{i-1}p_i$ intersects with line $p_{i+2}p_{i+1}$, but all edges which take the remaining *reflex* vertices of P as endpoint don't intersect with ray line $p_{j+1}p_j$(the endpoint p_{j+1}) or ray line $p_{i-1}p_i$ (the endpoint p_{i-1}) (See Fig. 1(d)).

After confirming whether the conditional propositions 1~4 are true or not respectively, and the case 3 can be divide into the following (i)~ (iv) for partitioning P.

(i) If both the proposition 1 and 2 are true, then according to property 1, p_ip_j is the subdivision line segment of P and both two *reflex* vertices p_i and p_j are removed;

(ii) if both the proposition 1 and 3 are true, then according to the property 1, p_ip_j is the subdivision line segment of P and one *reflex* vertex is removed;

(iii) if the proposition 1 is false but the proposition 4 is true, then according to property 1, p_iq_i or p_jq_i is the subdivision line segment of P and one *reflex* vertex is removed, where $q_i(x_i,y_i)$ is the intersection of ray line $p_{j+1}p_j$ (or $p_{i-1}p_i$);

(iv) if both the proposition 1 and 4 are false, then starting from p_j find its next *reflex* vertices, according to the cases 1-3 to handle and obtain its subdivision line segment.

IPSPCD module can be described as follows:

Input: Polygon $P(p_1,p_2,\cdots,p_n)$, whose vertices are ordered in clockwise order;

Output: the resulting convex polygons;

Step1. Judge and mark the *reflex* vertices of P through rule 1, count the number N of its *reflex* vertices, l=1, k=N ;

Step2. If k=0, then output P, stop partition; otherwise, if k=1, then find its *reflex* vertex $p_i(1\leqslant i\leqslant n)$ and compute its subdivision point p_j as case 1 and output $P_l=\{p_i, p_{i+1},\cdots, p_j,p_i\}$, $P_{l+1}= \{p_1,\cdots,p_i,p_j,\cdots,p_n\}$, stop partition, otherwise, i=1, go to next step;

Step3. Starting from p_i, in clockwise order, find the next *reflex* vertex p_j (when j=N, then starting form p_1 again). If p_i and p_j belong to the case 2, output P_l={p_i,p_{i+1} and q_i are ordered in clockwise}, set $P=\{p_1,\cdots, p_{i-1}, q_i\ p_{i+1}, \cdots,p_n\}$ and k=k-1; otherwise, go to next step;

Step4. if p_i and p_j belong to the case 3(i or ii), output $P_l=\{p_i,p_{i+1},\cdots,p_j,p_i\}$,set $P=\{p_1,\cdots,p_i, p_j,\cdots,p_n\}$ and k=k-2 for case 3(i)(or k=k-1 for case 3(ii)); otherwise, if it belongs to the case 3(iii), then output $P_l=\{p_i, q_i,\cdots,p_{i-2}, p_{i-1}\}$,set $P=\{p_1,\cdots,p_i, q_i,\cdots,p_n\}$ and k=k-1. Go to next step;

Step5. Set i=j, l=l+1, if k<2, then go to step2; otherwise, go to step3;

Given an irregular polygon with N *reflex* vertices, our IPSPCD method uses N+1 single list structure to store the N+1 convex polygons resulting. The number of its storage units is the same as [11], but less n+2N storage units of weigh than [15]. Our IPSPCD method doesn't need to code and calculate weight, so its time complexity is $O(n+N)$, lower than [10] [11][15]. Experiments show the number of convex polygons decomposed by our IPSPCD method is the same as that done by [10][11] and smaller than or equal to that done by [9] (see appendix B).

III. CLIPPING BETWEEN TWO CONVEX POLYGONS

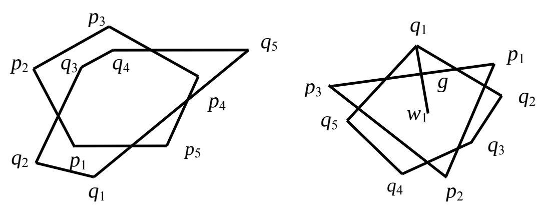
After minimum convex decompositions for irregular polygons are implemented, quickly clipping convex polygons against convex polygon window is a key step for improving computation efficiency of their overlapping area.

Some classical algorithms, concerning the clipping against polygon windows, have been published such as Sutherland-Hodgman algorithm[17] and Cyrus-Beck[18] whose cutting window are rectangle and convex polygon respectively. In addition, Weiler et al[6] present an clipping algorithm for eliminating the hidden surfaces and line segments. In 2003, Huang You-qun et al[8] put forward a fast clipping algorithm in which clipping window is a non convex polygon. For clipping algorithm in [9](2007), resulting convex polygons are organized in a binary space partition tree. By using it, the candidate convex polygons, which intersect with clipped line segment, are found quickly. In this section, on basis of fast intersection test between the line segment and the convex broken line segment, we suggest the fast convex polygon clip method (FCPC method) for CIPOA algorithm.

Definition 4: Let P_l and Q_l be two polygons and p be the vertex of P_l , if p lie in Q_l, then the attribute of p_l about Q_l is called as "internal"(value:1), otherwise as "external" (value:0), and directed edge with start point (attribute:1) and endpoint (attribute:0) is called as 1-0 edge of P_l about Q_l. Likewise 0-1,0-0 and 1-1 edge of P_l about Q_l can be defined.

Property 3: Judging the position relation (namely internal or external of definition 4) of a point with any polygon through intersection parity test (IPT) method, its complex operation is $O(1)$ multiplication-divisions (with regard to IPT method and proof of this property, see appendix A).

Thought of Method Suppose that P_l and Q_l are convex polygon with $n1$ vertices and one with $m1$ vertices, we use [3] to detect whether they intersect or not. If not, then $P_l\cap Q_l=\Phi$(null), otherwise, return their common point v_1. Then using IPT method, we calculate the internal and external attribute values of all vertices of Q_l about P_l and P_l about Q_l quickly, and count the numbers $c1$, $c2$ of internal vertices of P_l and Q_l respectively. (i) If $c1$=$n1$(or $c2$=$m1$), then $P_l\cap Q_l = P_l$ (or Q_l); (ii) If 0<$c1$<$n1$(Fig. 2(a))(if $c1$= 0 and 0<$c2$<$m1$, then P_l and Q_l are interchanged), then we determine all intersections of P_l and Q_l quickly. Its process is as follows: Construct edge list of P_l: $S_p=\{e_{p,1},e_{p,2},\cdots,e_{p,n1} \mid e_{p,1}:1\text{-}0\}$ and that of Q_l: $S_q=\{e_{q,1}, e_{q,2},\cdots, e_{q,m1}\}$. Starting from $e_{q,1}$, we search for $e_{q,v}$ which intersects with 1-0 edge $e_{p,1}$, and calculate their intersection; the next process is divided into two cases: (a) when $e_{q,v}$ ($1\leqslant v\leqslant m1$) is 0-1 edge, we search for 1-0 edge $e_{q,v+i}$ of S_q in clockwise order, then starting from the next edge of $e_{p,1}$, search for $e_{p,u}$(1⩽



(a) (b)

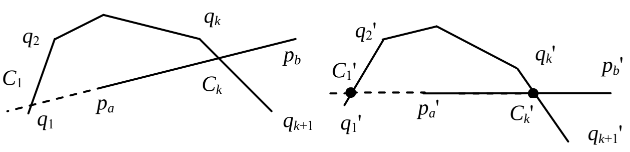
(a) At least an internal vertex of P_l about Q_l (b) No internal vertex

Fig. 2 the two cases of intersection

$u \leqslant n1$) which intersects with $e_{q,v+i}$, ⋯; (b) when $e_{q,v}$ is 0-0 edge, starting from the next edge of $e_{p,1}$, we search for the edge $e_{p,u}$ of S_p which intersects with $e_{q,v}$,⋯. Continuously, the next 1-0 edge of S_p or S_q is met or all 1-0 edge in S_p and S_q are traversed. For the former, repeat above step until all their intersections are computed. (iii) If $c1$=0 and $c2$=0 (see Fig. 2(b), then we calculate the intersection g of the line segment w_1q_1 and P_l and regard g as an internal vertex of P_l about Q_l. So, it is reduced to (ii).

In addition, for a line segment and any convex broken line segment, we give shear-transformation based intersection computation (STIC) method. In our FCPC method, it can be applied to determine quickly the intersections of P_l and Q_l. STIC method is described as follows:

It is shown as Fig. 3(a) that p_ap_b is a line segment with endpoint p_a (x_a,y_a) and p_b (x_b,y_b), the broken line segment consists of k line segment q_1q_2, q_2q_3,⋯,q_kq_{k+1}. Where the two coordinates of q_i are x_i and y_i respectively (i=1,⋯,k). If $x_a \neq x_b$ and $y_a \neq y_b$, then through translation and shear transformation given by (2), p_ap_b and q_iq_{i+1} (i=1,⋯,k) are



(a) Before transform (b) After transform

Fig. 3 The intersecting test and computing intersection based shear transformation

$$T = \begin{bmatrix} 1 & d \\ 0 & 1 \\ -x_a & -y_a \end{bmatrix}, \left(d = \frac{y_a - y_b}{x_b - x_a} \right) \quad (2)$$

transformed into the horizontal line segment $p_a'p_b'$ and the line segment $q_i'q_{i+1}'$ with endpoint $q_i'(x_i',y_i')$ and $q_{i+1}'(x_{i+1}', y_{i+1}')$(Fig.3(b)). Where $y_i'=d(x_i-x_a)+y_i-y_a$, $y_{i+1}'=d(x_{i+1}-x_a)+y_{i+1}-y_a$. If the signs of y_i' and y_{i+1}' are the same or the abscissa $x_{c,i}'$ of the intersection C_i' between the line $q_i'q_{i+1}'$ and $p_a'p_b'$ doesn't satisfy $x_i' \leqslant x_{c,i}' \leqslant x_{i+1}'$, then q_iq_{i+1} and p_ap_b don't intersect; Otherwise, their intersection is $C_i(x_{c,i}' + x_a, y_a - dx_{c,i}')$. Where $x_{c,i}'$is computed by (3).

$$x_{c,i}' = x_i - x_a - \frac{(x_{i+1} - x_i)y_i'}{y_{i+1}' - y_i'} \quad (3)$$

Note: STIC method also can be used to the intersection test and the computation of a line segment and any non-convex broken line segment. Its two special cases are that (a) if k=1,then it tests on two line segments; (b) if changes the line segment the ray line, then $x_i' \leqslant x_{c,i}' \leqslant x_{i+1}'$ is replaced into $x_i' \leqslant x_{c,i}'$ or $x_{i+1}' \leqslant x_{c,i}'$, if the line, then it is removed.

FCPC module can be described as follows:

Input: Vertices of P_l: $p_1,p_2,\cdots,p_{n1}$ and vertices of Q_l: $q_1, q_2,\cdots,q_{m1}$ in clockwise order;

Output: A convex polygon P_c in clipping list L;

Step1. Use algorithm [3] to detect whether P_l and Q_l intersect or not. If not, then output null, stop clipping; otherwise, attain one common point v_1 of them, go to next step;

Step2. Use IPT method to determine the attribute value of p_1~p_{n1} about Q_l and that of q_1~q_{m1} about P_l. Count the numbers $c1$ and $c2$ of P_l and Q_l respectively. If $c1=n1$(or $c2=m1$), then L<=P_l (or Q_l), output P_c in L, stop clipping, otherwise, construct S_p and S_q , go to next step;

Step3. If $c1+c2=0$ calculate the intersection g of w_1q_1 and P_l as a internal vertex of P_l, go to next step;

Step4. Starting from $e_{q,1}$ of S_q, use STIC to find $e_{q,i}$ that intersects with $e_{p,1}$ of S_p and calculate their intersection. Store it into L and move $e_{q,1}$,⋯,$e_{q,j-1}$ to the tail of S_q in turn. Set i=2, go to Step6;

Step5. If $e_{p,i}$ is 1-0 edge, then starting from $e_{q,j+1}$ of S_q, use STIC method to find $e_{q,j+v}$ that intersects with $e_{p,i}$, calculate their intersection and insert it into the tail of L. Set j=j+v, go to next step;

Step6. If $e_{q,j}$ is 0-1 edge, then in clockwise order of Q_l insert its internal vertices from $e_{q,j}$ to the next 1-0 edge $e_{q,j+v}$ into the tail of L in turn. Starting from $e_{p,i}$ of S_p, use STIC method to find $e_{p,i+u}$ which intersects with $e_{q,j+v}$ and calculate their intersection. Insert it into the tail of L. Set i=i+u, j=j+v,If $e_{p,i}$ is 0-1edge, then go to Step7; If $e_{p,i}$ is 0-0 edge, go to step 8;

Step7. In turn insert all internal vertices between $e_{p,i}$ and next 1-0 edge $e_{p,i+u}$ of P_l into the tail of L in clockwise order, if $e_{p,i+u}$ is adjacent to $e_{p,1}$, output P_c in L, stop clipping; otherwise set i=i+u, go to step 5.

Step8. Starting from $e_{q,j+1}$ Use STIC method to find $e_{q,j+v}$ that intersected with $e_{p,i}$ and calculate their intersection. Insert it into the tail of L, set j=j+v, if j=m1, then output P_c in L, stop clipping; otherwise, if $e_{q,j}$ is 0-1 edge, go to step 6; otherwise, go to next step;

Step9. Starting from $e_{p,i+1}$, use STIC to find $e_{p,i+u}$ that intersects with $e_{q,j}$, calculate their intersection and insert it into the tail of L. Set u=u+i, v=v+j, if $e_{p,i}$ is 0-1 edge, go to step 7; otherwise, go to step 8;

The time complexity of [18] is $O(n1m1)$. According to property 3, the time complexity of step1 in our FCPC method is $O(n1+m1)$. It is known from [3] that the time complexity of its step3 is $O(\log(n1)+\log(m1))$. So, we can deduce that the time complexity of its step4-7 is $O(n1+m1)$. And then the time complexity of Our FCPC method is $O(n1+m1)$, which is lower than [18]. Because [8][9] are non-convex clipping algorithm, the comparison for our CIPOA algorithm and them is only given in Section IV.

IV. Calculating the Overlapping Area of Two Irregular Polygons

After computing the vertices (x_i,y_i) (i=1,⋯,n2) of overlapping region for each pair of convex polygons to decomposition through FCPC method, its area can be calculated by (4).

$$S = \frac{1}{2}\sum_{i=1}^{n2}\left(x_i y_{i+1} - x_{i+1} y_i\right) \quad (4)$$

CIPOA algorithm can be described as follows:

Input: $P = \{p_1, p_2,\cdots, p_n\}$, $Q = \{q_1, q_2,\cdots,q_m\}$ in clockwise order;

Output: the area S of $P \cap Q$;

Step1. Using IPSPCD module to decompose P and Q into $P = P_1 \cup \cdots \cup P_{n2}$, $Q = Q_1 \cup \cdots \cup Q_{m2}$, Set S=0, i=1 go to next step;

Step2. If $i > n2$ then output S, the calculation end, otherwise Set j=1 go to next step;

Step3. Using FCPC module to solve $P_i \cap Q_j$ and calculate its area S_{ij} by (4) go to next step;

Step4. $S = S + S_{ij}$, If $j<m2$, then $j = j+1$, go to Step3, otherwise $i = i+1$, go to Step2;

V. Examples and Algorithm Analysis

A. Examples

Experiment environment: PC with 2.4G, 1G memory, VC + + 6.0.

Example1 it cites from [3], Fig. 4 shows two irregular polygons: P={(7,2),(4,2),(3,3),(0,4),(1,8),(3,8),(4,6),(5,6),(6,8),(9,5)};Q={(7,0),(2,1),(8,3),(7,9),(11,7),(12,5),(10,1)} . The algorithms in [4][5] and our CIPOA can all compute the overlapping area and its value is 2.99. The computing time is shown in Table I.

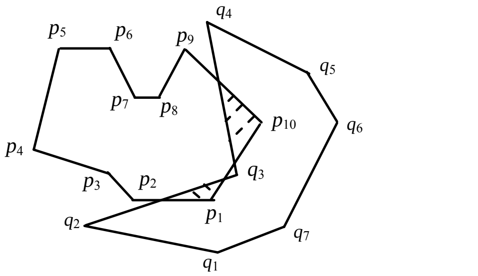


Fig. 4 The test example 1 for calculation of overlapping area

Example2 Fig. 5 shows two irregular polygons: P={(0.5, -1),(-1.5,-4),(-2,3),(-1,1),(1,1),(1.5,3),(3,2), (2, 0.5),(3.5,-1), (2.5,-2),(2,-2),(1,-1)}; Q={(0,-3),(-4,-3),(-1,-1),(-3,2.5),(1, 2),(3,4), (3,1)}. The algorithms in [4][5] and Our CIPOA can all compute the overlapping area and its value is 13.06. The computing time is shown in Table I.

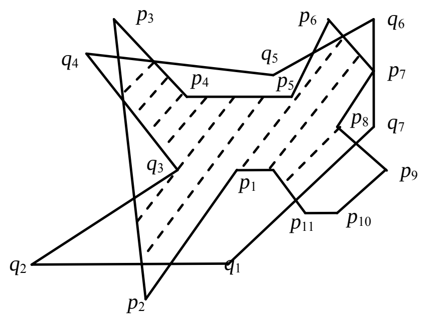


Fig. 5 The test example 2 for calculation of overlapping area

Example3 (a) Fig.6 shows two irregular polygons: P= {(2, 2),(0,2),(2,5),(3.5,3.5),(5,3.5),(7,5),(8,3.5),(5.5,2),(6,-1),(0,-1),}; Q={(1.5,3), (0,4), (1,5), (2,4), (3.5,4), (6,5), (7, 2.5),(5,3),(4,0),(3,0),(2,2.5),(1,0),(0,1)};

(b)P=(9,2),(7,2), (9,5),(10.5,3.5),(12,3.5),(14,5),(15,3.5), (12.5,2),(13,-1),(7, -1),};Q={(8.5,3),(7,4),(8,5),(9,4), (10.5,4),(13,5),(14, 2.5), (12,3),(11,0),(10,0),(9,2.5),(8,0), (7,1)}. All algorithms in [4] [5] and Our CIPOA can compute the overlapping area in (a) and (b) and their values are 12.32 and 0, respect-tively. The computing time is shown in Table I.

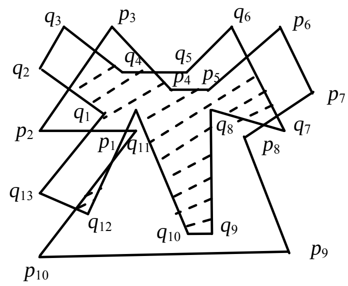


Fig.6 The test example 3(a) for calculation of overlapping area

TABLE I
COMPUTING TIME COMPARISON FOR THREE ALGORITHMS(ms).

Example No.		Cost time	Ref [4]	Ref [5]	Ref [8]	Ref [9]	Our CIPOA
1		Partition	0.031	0.015		0.015	0.010
		clipping	0.266	0.266	0.248	0.193	0.103
		Total	0.296	0.281	0.248	0.208	0.113
2		Partition	0.047	0.016		0.025	0.015
		clipping	0.391	0.438	0.286	0.311	0.152
		Total	0.437	0.454	0.286	0.336	0.167
3	(a)	Partition	0.078	0.016		0.026	0.015
		clipping	0.687	0.328	0.319	0.327	0.261
		Total	0.766	0.359	0.319	0.353	0.276
	(b)	clipping	0.650	0.616	0.271	0.201	0.180
		Total	0.750	0.648	0.271	0.227	0.196

B. Algorithm Analysis

Let N and M denote the numbers of *reflex* vertices of two irregular polygons P with n vertices and Q with m vertices respectively, where, $N<<n$, $M<<m$. The link list is used in Our CIPOA algorithm. The storage space it needs is the same as [7][8] but less than [6][9](data in [6] [9] are all organized into tree structure). The time complexity of [9] is $O(m\log n)$ in most cases, but $O(mn)$ in the worst case. The time complexity of [8] is $O(mn)$, because n sear transformations are executed on the polygon window Q when n edges of P are transformed to a horizontal line in turn. After P and Q are decomposed through our IPSPDA module, the numbers of their convex polygons can be at most N+1 and M+1, respectively. That is to say, our CIPOA needs to clip (N+1)(M+1) pairs of convex polygons at most. We can draw a conclusion that the time complexity of the clipping component of CIPOA is $O(Mn+Nm)$ which is lower than that of [8][9]. Furthermore, the time complexity of our CIPOA algorithm is also $O(Mn+Nm)$, but both time complexity of [4][5] are $O(mn)$ which is higher than that of our CIPOA. Therefore the computational efficiency of our CIPOA algorithm is superior to [4][5][8][9] for computing overlapping area of two irregular polygons. The experiment data in table I prove it.

C. Algorithm application for evolutionay polygon layout

Our CIPOA algorithm consists of IPSPCD module, FCPC module and the component of computing overlapping area. The contribution of IPSPCD module to interference calculation in evolutionary polygon layout lies in its minimum convex polygons to the decomposition, which plays an important role for improving efficiency of FCPC module. This is because its time complexity $O(Mn +Nm)$ of each iteration has a linear relation with the numbers of convex polygons to their two decompositions.

In the process of evolution, its implementation is at either first iterative or pro-process stage. So its influence to efficiency of CIPOA algorithm and the evolutionary polygon layout algorithm can be omitted. For evolutionary polygon layout, FCPC module is executed at each iterative step, therefore, it plays the key role for improveing the calculation efficiency of our CIPOA algorithm and evolutionary polygon layout algorithm. For further improving speed of the interference calculation we can combine FCPC method with 2D axis-aligned rectangle bounding box test in evolutionary polygon layout.

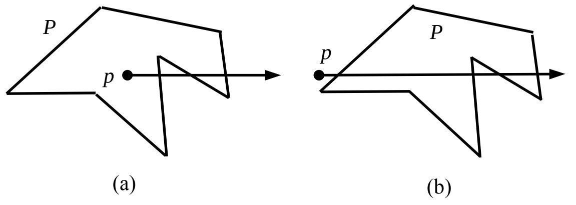
VI. CONCLUSION

In this paper, an efficient algorithm for computing the overlapping area between two irregular polygons is developed. It is based on decomposing irregular polygon into the minimum number of convex polygons and clipping convex polygon against a convex polygon. Numerical experiments show that calculation efficiency of our CIPOA algorithm outperforms the existing algorithms, and the less two irregular polygons overlap, the more obvious computation efficiency of our CIPOA algorithm improves. It can be applied to compute the overlapping area for both the convex and the non-convex polygon objects. Therefore, our CIPOA algorithm can be suitable for calculating their overlapping area for the evolutionary polygon layout. Its validity for other problems should be discussed in the future.

APPENDIX A IPT METHOD AND ITS TIME COMPLEXITY

It is shown as Appendix Fig. 1(a)(b) in which gives a point p and any polygon P, IPT method is described as follows:

Taking p as the endpoint draw the horizontal ray line, we in turn detect whether each edge of P intersect with it or not and count the number of their intersections. If the number is odd number (see Fig. (a)), then p lies in interior of P; otherwise, p lies in exterior of P(see Fig. (b).



Appendix Fig.1 The relation between a point p with any polygon P. (a) p lies in interior of P, (b) p lies in exterior of P.

In the process of counting intersection, there are two special intersecting cases as follows:

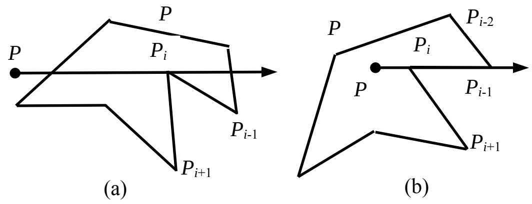
(a) For an edge of P, if an endpoint of the edge lie on the horizontal ray line we draw, but another one lies below it, then we consider that they don't intersect.

For example, in Fig. (a), both two edges $p_{i-1}p_i$ and p_ip_{i+1} of P don't intersect with the horizontal ray line.

(b) For an edge of P, if it coincides with the horizontal ray line we draw, then consider that they don't intersect.

For example, in Fig. (b), the edge $p_{i-1}p_i$ of P doesn't intersect with the horizontal ray line we draw.

Let P be a planar polygon with N *flex* vertices, $p_i(x_i,y_i)$ (i=1,2,…,n) be its n vertices of P, $p(x_d,y_d)$ be a point on



App. Fig.2. Two special cases of intersection of the horizontal ray line and P. (a) Passes through a vertex of P; (b) Coincides with an edge of P.

the plane then the x coordinate I_x of the intersection can computed by (1) and procedure module of IPT method is described as follows:

for i=1, 2,…,n
    if (($y_i > y_d$ and $y_{i+1} \leqslant y_d$) or($y_{i+1} > y_d$ and $y_i \leqslant y_d$))
        if($I_x \geqslant x_d$) j=j+1;
if (j%2=0) the attribute (p)=1;
else the attribute (p)=0;

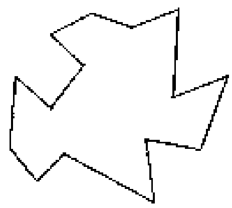
$$I_x = x_i + \frac{y_d - y_i}{y_{i+1} - y_i}(x_{i+1} - x_i) \quad (1)$$

Proof of the property 3:

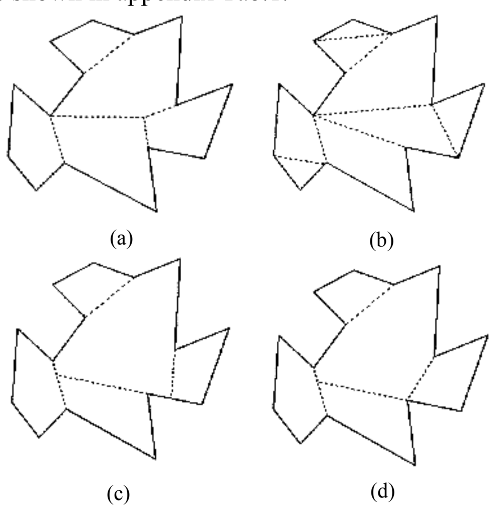
For the polygon with N *flex* vertices it is obvious that the number of the edges of P intersect with the horizontal ray line we draw is 2(N+1) at most. So its complex operation is 4(N+1) multiplication-divisions at most, namely O(1) multiplication-divisions. Particularly, when P is a convex polygon (N=0), its complex operation is 4 multiplication-divisions at most.

APPENDIX B COMPARISON OF 4COMPOSITION ALGORITHM

Appendix example1. Appendix Fig. 3 cites from [16] (pp.54). Their graphs de-composed by [10][9][11] and our IPSPCD method are shown as (a), (b), (c) and (d) in appendix Fig. 2, respectively and the numbers of decompositi-on are shown in appendix Tab.1.

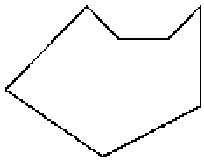


Appendix Fig.3 The test graph 1 for comparison of 4 algorithms.

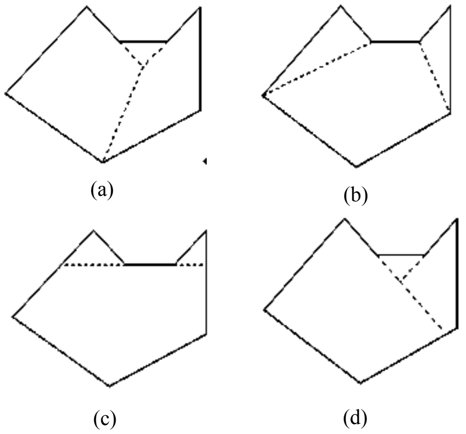


Appendix Fig.4 Partition graphs of 4 algorithms for the test graph 1. (a) Partition graph for [10]; (b) Partition graph for [9]; (c) Partition graph for [11]; (d) Partition graph for our IPSPCD method.

Appendix example2. In appendix Fig. 5, P={(0,-3), (-6, 1),(-1,6), (1,4), (4,4), (6, 6), (6,0)} is a polygon with 2 reflexes. The graphs decomposed by [10][9][11] and our IPSPCD method are shown as (a), (b), (c) and (d) in appendix Fig.4, respectively, and the numbers of convex polygons to partition are shown in appendix Tab.1.

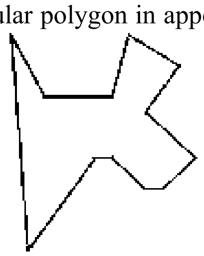


Appendix Fig.5 The test graph 2 for comparison of 4 algorithms.

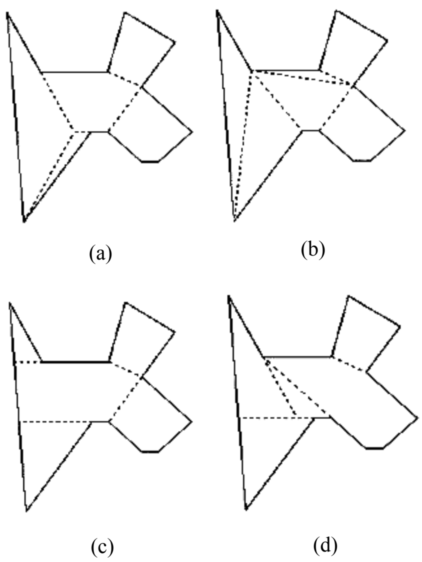


Appendix Fig.6 Partition graphs of 4 algorithms for the test graph 2. (a) Partition graph for [10]; (b) Partition graph for [9]; (c) Partition graph for [11]; (d) Partition graph for our IPSPCD method.

Appendix example3. An irregular polygon in appendix Fig.7 cites from example 2 in Section V. The graphs decomposed by [10][9][11] and our IPSPCD method are shown as (a),(b),(c) and (d) in appendix Fig. 3, respectively, and the numbers of convex polygons to partition are shown in appendix Tab.1.

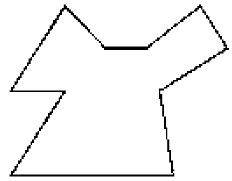


Appendix Fig.7 The test graph 3 for comparison of 4 algorithms.

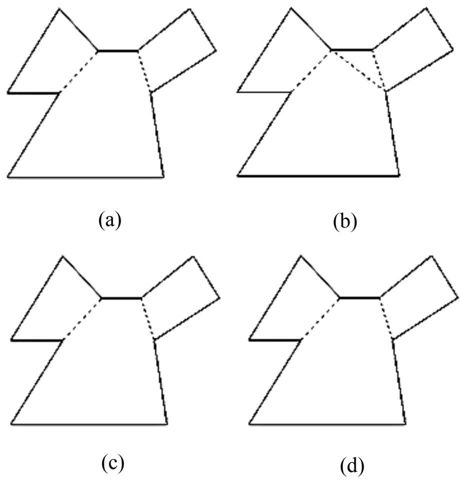


Appendix Fig.8 Partition graphs of 4 algorithms for the test graph 3. (a) Partition graph for [10]; (b) Partition graph for [9]; (c) Partition graph for [11]; (d) Partition graph for our IPSPCD method.

Appendix example4. An irregular polygon in appendix Fig. 9 takes from example 3 in section V. Their graphs decomposed by [10] [9][11] and our IPSPCD method are shown as (a), (b), (c) and (d) in appen-dix Fig.10, respectively, and the numbers of de-composition are shown in appendix Tab.1.



Appendix Fig.9 The test graph 4 for comparison of 4 algorithms.



Appendix Fig.10 Partition graphs of 4 algorithms for the test graph 4. (a) Partition graph for [10]; (b) Partition graph for [9]; (c) Partition graph for [11]; (d) Partition graph for our IPSPCD method.

APPENDIX TABLE I
COMPARISON OF THE NUMBERS OF CONVEX POLYGONS TO A PARTITION FOR 4 ALGORITHMS

Appendix example Number	Ref. [10]	Ref. [9]	Ref. [11]	Our IPSPCD
1	5	8	5	5
2	3	3	3	3
3	5	6	5	5
4	3	4	3	3

ACKNOWLEDGMENT

This work was supported in part by a grant from the Hunan Provincial Natural Science Foundation of China (11JJ6050) and Research Foundation of Education Bureau of Hunan Province, China (11A120) and the Doctor Startup Foundation of Xiangtan University of China (Grant No. 09QDZ18)

REFERENCES

[1] J. J. Jiménez, R. J. Segura, “Collision detection between complex polyhedra,” Computer Graphics, vol. 32, Feb. 2008, pp. 402-411.
[2] Bennell J A, Dowsland W B, “The irregular cutting-stock problem: a new procedure for deriving the no-fit polygon,” Computers and Operations Research, vol. 28, Mar. 2001, pp.271-287.
[3] Chazelle B. M, Dobkin D. P. “Detection is easier than computation. Proc,”12th ACM Annual Sympos. on Theory

and Comput.,Princeton University Press, May,1980, pp. 146-153.
[4] Liu De-quan, Teng Hong-fei, "Finite triangular element method of computing overlapping area between two irregular-shaped polygons," Chinese Journal of Dalian University of Technology, vol. 39, Sep. 1999, pp. 597-600.
[5] Wang Jin-min, Liu Ji-ye, Fang Yi, "Algorithm of computing overlapping area between two irregular-shaped polygons based on trapezium decomposition," Chinese Journal of Engineering Graphics, Jun. 2005, pp. 52-57.
[6] Weiler K, Atherton P, "Hidden surface removal using polygon area sorting," In: Proceedings of the SIGGRAPH'77. New York: ACM Press, 1977, pp.214-222.
[7] Greiner G, Hormann K, "Efficient clipping of arbitrary polygons," ACM Transactions on Graphics, vol. 17, Feb. 1998, pp. 71-83.
[8] Liu Yong-kui, Gao Yun, Huang You-qun, "An efficient polygon clipping algorithm," Chinese Journal of Software, vol. 14, Apr. 2003, pp. 845-856.
[9] Li Jing, Wang Wen-cheng, Wu En-hua, "Line Clipping Against a Polygon Based on Convex decomposition," Chinese Journal of Computer-Aided Design & Computer Graphics, vol. 19, Apr. 2007, pp. 425-429.
[10] Chazelle B, Dobk in PD, "Optimal convex decompositions. In: Toussaint T Godfrieded," Computational Geometry, Amsterdam, Holland: North-Holland, 1985, pp.63-133.
[11] Xiao Zhong-hui, Lu Zhen-rong, Zhang Qian, "Coding algorithm for decomposition a simple polygon into convex parts," Chinese Journal of Computers, vol. 19, Jun. 1996, pp.477-481.
[12] Keil J. M. "Decomposing a polygon into simpler components," SIAM J. Comput., vol. 14, Apr. 1985, pp. 799-817.
[13] Keil J. M, Sack J.-R. "Minimum decomposition of polygonal objects," In G. T. Toussaint, editor, Computational Geometry, North-Holland, Amsterdam, Netherlands, 1985, pp. 197-216.
[14] Keil M, Snoeyink. "On the time bound for convex composition of simple polygon," International Journal of Computational Geometry and Application, vol. 12, Mar, 2002, pp. 181-192.
[15] Wang Zheng-xuan, Li Wen-hui, Pang Yun-jie, "A weighting algorithm for decomposing a simple polygon into convex polygons," Chinese Journal of Computers, vol, 21, Mar. 1998, pp. 229-233.
[16] De Berg Mark, Overmars Mark. "Computational geometry: algorithms and applications," 2nd ed. Berlin: Springer, 2000, pp.45-55.
[17] Sutherland I E, Hodgman G W, "Reentrant polygon clipping," CACM, vol. 17, Jan. 1974, pp. 32-42
Cyrus M, Beck J. "Generated two-and three-dimentional clipping," Computing Graphics, vol 3, Jan, 1978, pp. 23-28.

RESEARCH BACKGROUND

The research background of this paper is the National Natural Science Foundation of Hunan Provincial Natural Science Foundation of China (11JJ6050) and Research Foundation of Education Bureau of Hunan Province, China (11A120) and China with grant NO.61040009, 60773047 and the Doctor Startup Foundation of Xiangtan University of China with grant NO.09QDZ18. The layout problem belongs to combinatorial optimization problem with combinatorial explosion. In process of solving 2-D layout problem by using evolutionary algorithms, fast and accurate interference computation is necessary to speed up solving efficiency and quality, besides effective initial layout pattern and first-rate layout evolutionary algorithms.

In the past few years, much research has been done in the field. The achievements of our work have been published by some important domestic and oversea publications or proceedings, such as Chinese Journal of Computers, Journal of Computer-aided Design& Computer Graphics, Journal of Dalian University of Technology, IEEE Proceedings of 6th World Congress on Intelligent Control and Automation and 3rd International Conference on Intelligent Information Technology Application (IITA09).

This paper proposes an approach for computing overlapping area of irregular polygons. The calculating efficiency of our approach outperforms the existing algorithms.

Zi-qiang Li was born in Xiangtan, Hunan, China, on Jun. 24, 1963 and received the M.S. degree in computed mathematics from Dalian Technical University Dalian, China, in 1995 and the Ph.D. degree in computer applica-tion technology from Dalian Technical University, Dalian, China, in 2008, respectively. Major field of study includes computer graphics, computer vision and intelligent computation.

From 1995 to 2000, he was on the faculty of the Department of Computer Science and Technology, Xiangtan University, China. From 2001 to 2006, he was an associate professor of School of Information and Engineering, Xiangtan University, China. Since 2007, he has been a professor of school of information and engineering, Xiangtan University, China.

Dr. Li is a member of the Computer Society of the Republic of China. His research work obtained 1 project from the National Natural Science Foundation of China. Publishes are more than 20.

Yan He received the B.S. degree in computer science and technology from Xiangtan University, Xiangtan, Hunan, China, in 2008 and received the M.S. degree in computer science and technology from Xiangtan University, Xiangtan, Hunan, China, in 2011..

Her research interests include computer graphics and intelligent computation.

Zhu-jun Tian born in 1987, M.S candidate. His research interests include intelligent computation and computer graphics.