Polygon placement under translation and rotation

Jean-Daniel Boissonnat, Francis Avnaim

INRIA

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tel. (1) 39 63 55 11

Rapports de Recherche

N°889

POLYGON PLACEMENT UNDER TRANSLATION AND ROTATION

Francis AVNAIM
Jean Daniel BOISSONNAT

AOUT 1988

RR-0889

Polygon placement under translation and rotation
Placement de formes polygonales:2. Placement en translation et rotation

Francis Avnaim and Jean Daniel Boissonnat

INRIA Sophia-Antipolis
Route des Lucioles
06565 Valbonne, France

Abstract:
We present a general algorithm which computes an exact description of the set of all placements for a polygon I (with m edges) which is free to translate and/or to rotate but not to intersect another polygon E (with n edges). The time complexity of our algorithm is $O(m^3n^3 \log mn)$ which is close to optimal in the worst-case. Moreover, in some practical situations, the time complexity is only $O(n \log n)$. This algorithm is rather simple and has been implemented. It can be used as an efficient tool in several applications such as cutting stock, inspection and motion planning for a two dimensional robot admidst polygonal obstacles.

Résumé:
Cet article présente un algorithme général qui calcule une description analytique exacte de l'ensemble des placements d'un polygone I (ayant m arêtes) libre de se déplacer en translation et rotation dans le plan sans intersecter un polygone E (ayant n arêtes). La complexité de l'algorithme est de $O(m^3n^3 \log mn)$ ce qui est proche de l'optimal dans le cas le pire. On montre de plus que la complexité réelle de l'algorithme, dans certaines situations pratiques est $O(n \log n)$. L'algorithme présenté est assez simple et a été implanté. Il recouvre un champ d'applications varié incluant les problèmes de découpe automatique, de conformité de pièces industrielles ou les problèmes de planifications de trajectoires pour un robot mobile plan évoluant au milieu d'obstacles polygonaux.

1 Introduction

Given are two polygonal regions E and I, with n and m edges respectively. We want to find the closure of the subset $P(I,E) \subset [0,2\pi[\times R^2$ consisting of all free placements $(\theta,\vec{u})$ satisfying $T_{\vec{u}} \circ R_\theta(\partial I) \cap \partial E = \emptyset$. $T_{\vec{u}}$ denotes translation by vector $\vec{u}$, R_θ denotes rotation with center at the origin and angle θ, ∂I (resp. ∂E) denotes the boundary of I (resp. E).

This problem has several applications in Robotics and Computer Vision, and appears in problems such as cutting of plates out of a sheet of material [3], checking if a shape measured by a vision system satisfies some given tolerances [4] or computing the position of a robot from geometric features measured on the boundary of its (known) environment. Moreover, the above problem is intimitely related to the path planning problem [2,9,13,14]

This problem has been attacked from different points of view in the Computational Geometry literature and is the same, in essence, as the polygon containment problem and the computation of the convolution of two polygons. When only translations are allowed, several algorithms have been proposed for both problems [1,6,7,8]. When both translations and rotations are allowed, Chazelle [6] has proposed an $O(m^3n^3(m+n)\log mn)$ algorithm that computes all stable free placements of I inside E, i.e. in which I makes three contacts with the boundary of E simultaneously. When both I and E are convex, the time complexity is reduced to $O(mn^2)$. When only I is convex, Leven and Sharir [10] have proposed an algorithm which computes all stable free placements in $O(mn\lambda(mn)\log mn)$ time. Here $\lambda(q)$ is an almost linear function of q. Based on this result, Kedem and Sharir [9] have computed the topological and combinatorial structure of the 3-dimensional set of free placements within the same time bound. The general case where both I and E may be non convex has been solved by Schwartz and Sharir [13]. The complexity of their algorithm is $O(n^5)$ (m is considered fixed) in the case that I is a line segment (a *ladder*). This algorithm is rather involved and, according to the authors themselves, several technical delicate issues are ignored.

In this paper, we propose a rather simple method to handle the general case where both I and E may be non convex and even not connected. Our algorithm provides a complete description of the set of all free placements. Its time complexity is $O(m^3n^3\log mn)$ which is close to optimal in the worst case since the solution may consist of $\Omega(m^3n^3)$ faces. Moreover, in some practical situations, it is shown to be $O(n\log n)$. This algorithm is rather simple and has been implemented.

The paper is organized as follows. In Section 2, we introduce the notions of contact and of contact regions which are portions of the boundary of $P(I,E)$. In Section 3, we present an algorithm to compute a contact-region and in Section 4, we show how to compute $P(I,E)$. Final remarks are made in Section 5 and we conclude in Section 6.

2 Contacts and contact-regions

Let $I_1, \cdots, I_m$ be the vertices of I and $i_1 = [I_1, I_2], \cdots, i_m = [I_m, I_1]$ its edges. Similarly let $E_1, \cdots, E_n$ and $e_1, \cdots, e_n$ denote the vertices and the edges of E. Both I and E are defined in a reference frame $\mathcal{R}_0 = (O, x, y)$ and we will identify, in the sequel, vector $\vec{u}$ and the point M satisfying $\overrightarrow{OM} = \vec{u}$. For any subset $\mathcal{A}$ of the plane, we note $\mathcal{A}^*$ the subset deduced from $\mathcal{A}$ by symetry of center O. For any point M, we note M_θ the point $R_\theta(M)$ and for any line segment a we note a_θ the line segment $R_\theta(a)$.

We define a *vertex-edge contact* between I and E as a pair (I_l, e_s). Similarly, we define an *edge-vertex contact* to be a pair (E_s, i_l).

We define a *contact-placement* associated to contact (I_l, e_s) (resp. (i_l, E_s)) to be a placement $(\theta, \vec{u})$ satisfying : $T_{\vec{u}} \circ R_\theta(I_l) \in e_s$ (resp. $E_s \in T_{\vec{u}} \circ R_\theta(i_l)$).

A contact-placement is said to be *free* if, in addition, it satisfies : $\forall j, k : T_{\vec{u}} \circ R_\theta(i_j) \cap e_k = \emptyset$. An example of a free contact-placement is shown in Figure 1.

We define the *contact-region* associated to a contact (I_l, e_s) (resp. (i_l, E_s)) to be the closure of the set of all free contact-placements associated to contact (I_l, e_s) (resp. (i_l, E_s)), and denote it $R(I_l, e_s)$ (resp., $R(i_l, E_s)$).

A *double-contact-placement* (or *double-contact* for short) is a (non necessarily free) contact-placement associated to two distinct contact-placements.

A *triple-contact-placement* (or *triple-contact* for short) is a (non necessarily free) contact-placement associated to three distinct contact-placements.

Notice that a contact region $R(C)$ may be empty. This means that for all positions of I involving contact C, ∂I intersects ∂E. The relevance of the notions of contact-placement and contact-region comes from the fact that they yield a discrete description of the boundary of $P(I, E)$, as stated by the following crucial, though trivial, proposition :

Proposition 1 $\partial P(I, E) = \bigcup_C R(C)$

In order to compute $\partial P(I, E)$ we will compute $R(C)$ for each contact C. To this aim, we need a more tractable characterization of $R(C)$.

Let us define, for a fixed orientation θ, $R_\theta(C)$ as the set of vectors $\vec{u}$ such that $(\theta, \vec{u}) \in R(C)$. It is clear that $R(C) = \bigcup_\theta R_\theta(C)$. Let $C = (J, f)$ be a contact. For any fixed orientation θ, $R_\theta(C)$ is the set of vectors $\vec{u}$ satisfying :

Condition 1 : Either $T_{\vec{u}}(J_\theta) \in f$ if C is a vertex-edge contact, or $J \in T_{\vec{u}}(f_\theta)$ if C is an edge-vertex contact.

Condition 2 : $\forall j, k : T_{\vec{u}}(i_{j_\theta}) \cap e_k = \emptyset$.

The set of vectors $\vec{u}$ satisfying Condition 1 is the line segment $T_{\overrightarrow{J_\theta O}}(f)$ if C is a vertex-edge contact and the line segment $T_{\overrightarrow{JO}}(f_\theta)^*$ if C is an edge-vertex contact. The set of vectors $\vec{u}$ not satisfying Condition 2 for given j and k is the interior of

a parallelogram $p_{jk}(\theta)$ (denoted $\dot{p}_{jk}(\theta)$). It is easy to see that $p_{jk}(\theta)$ is the parallelogram $(E_k, E_{k+1}, E_{k+1} - \overrightarrow{I_{j_\theta} I_{j+1_\theta}}, E_k - \overrightarrow{I_{j_\theta} I_{j+1_\theta}})$ translated by vector $\overrightarrow{I_{j_\theta} O}$ (see Figure 2). Thus for a vertex-edge contact we have:

$$R_\theta(J, f) = T_{\overrightarrow{J_\theta O}}(f) \setminus \bigcup_{jk} \dot{p}_{jk}(\theta) \tag{1}$$

and for an edge-vertex contact :

$$R_\theta(J, f) = T_{\overrightarrow{JO}}(f_\theta)^* \setminus \bigcup_{jk} \dot{p}_{jk}(\theta) \tag{2}$$

3 Computation of a contact-region

3.1 The case where only translation are allowed

In this section, we compute $R(J,f) = R_\theta(J,f)$ for a fixed value of θ. Let $f = AB$ and $\tilde{A}\tilde{B}$ denote either edge $T_{\overrightarrow{JO}}(AB)$ in case where C is a vertex-edge contact or edge $T_{\overrightarrow{JO}}(AB)^*$ in case where C is an edge-vertex contact. Equations (1) and (2) show that the boundary of the set of free placements consist of line segments which are portions of $\tilde{A}\tilde{B}$. Moreover, any contact-region $R(J,f)$ can be computed as stated by the following proposition :

Proposition 2 *$R(J,f)$ is a finite (possibily empty) family of segments which can be computed in time $O(mn \log mn)$.*

Proof : From Equations (1) and (2), $R(J,f)$ is the set theoretic difference between a segment and a family of (fixed) parallelograms. The intersection between a segment and a parallelogram is a segment (possibly reduced to a point) or the empty set and can be computed in constant time. Thus the set theoretic difference between a segment and a family of parallelograms is a family of segments. Computing a parallelogram p_{jk} takes constant time. As we have exactly mn parallelograms, computing all the $\{p_{jk}\}$ requires $O(mn)$ time. Each parallelogram p_{jk} has to be intersected with $\tilde{A}\tilde{B}$ yielding a family $\{]L_{jk}, M_{jk}[\}$ of segments in time $O(mn)$ (refer to Figure 3). Once the end points of these $O(mn)$ segments have been sorted along $\tilde{A}\tilde{B}$ (which takes $O(mn \log mn)$ time), $R(J,f)$ can be computed in time proportional to the number of these segments, which is $O(mn)$. This achieves the proof. □

Notice that the end-points of the segments composing $R(J,f)$ are free double-contacts.

3.2 The general case

Let $f = AB$ and $\tilde{A}\tilde{B}$ denote now either edge $T_{\overrightarrow{J_\theta O}}(AB)$ –in case that C is a vertex-edge contact– or edge $T_{\overrightarrow{JO}}(AB_\theta)^*$ –in case that C is an edge-vertex contact.

We have shown in Section 3.1 that, for any orientation θ, $R_\theta(C)$ is a (possibly empty) family of segments $S_1, \cdots, S_r$ which is the set theoretic difference between $\tilde{A}\tilde{B}$ and the union of parallelograms $\dot{p}_{jk}(\theta)$ (see the proof of Proposition 2 and Figure 3). Parallelogram $\dot{p}_{jk}(\theta)$ intersects $\tilde{A}\tilde{B}$ along a segment noted $]L_{jk}(\theta), M_{jk}(\theta)[$. Each end point of $S_i (i = 1, \cdots, r)$ is either a point $L_{jk}(\theta)$ or a point $M_{jk}(\theta)$ for some (j, k). To each segment $]L_{jk}(\theta), M_{jk}(\theta)[$, we associate the functions $\lambda_{jk}(\theta)$ and $\mu_{jk}(\theta)$, ranging in $[0, 1]$ and defined as follows :

$$\lambda_{jk}(\theta) = \frac{\|\overrightarrow{\tilde{A}L_{jk}(\theta)}\|}{\|\overrightarrow{\tilde{A}\tilde{B}}\|}$$

$$\mu_{jk}(\theta) = \frac{\|\overrightarrow{\tilde{A}M_{jk}(\theta)}\|}{\|\overrightarrow{\tilde{A}\tilde{B}}\|}$$

Notice that the family of segments $S_1, \cdots, S_r$ only depends on the order of the $\lambda_{jk}(\theta)$ and the $\mu_{jk}(\theta)$ on [0,1]. Moreover, $\lambda_{jk}(\theta)$ and $\mu_{jk}(\theta)$ are continuous functions of θ defined on sub-intervals of $[0, 2\pi[$. Thus their order on [0,1] changes only at a finite set of orientations, each one corresponding either to an intersection between two such functions or to an end point of a sub-interval defining such a function.

In order to describe $R(C)$, we will first compute the functions $\lambda_{jk}(\theta)$ and $\mu_{jk}(\theta)$.

3.2.1 Computation of $\lambda_{jk}(\theta)$ and $\mu_{jk}(\theta)$

We restrict our study to the case of a vertex-edge contact. The case of an edge-vertex contact can be easily reduced to the case of a vertex-edge contact by considering that E moves instead of I.

As the functions $\lambda_{jk}(\theta)$ and $\mu_{jk}(\theta)$ depend only on the orientation of I, we can choose J as the center of rotation R_θ. Thus the functions $\lambda_{jk}(\theta)$ and $\mu_{jk}(\theta)$ can be simply written :

$$\lambda_{jk}(\theta) = \frac{\|\overrightarrow{AL_{jk}(\theta)}\|}{\|\overrightarrow{AB}\|}$$

$$\mu_{jk}(\theta) = \frac{\|\overrightarrow{AM_{jk}(\theta)}\|}{\|\overrightarrow{AB}\|}$$

Let $p(\theta)$ be parallelogram $p_{jk}(\theta)$ for some (j, k). Parallelogram $p(\theta)$ is constructed on $T_{\vec{u}_\theta}(e_k)$ with vector $\vec{i}_\theta$. Here $\vec{u}_\theta$ (resp., $\vec{i}_\theta$) denotes the vector deduced from vector $\overrightarrow{I_jJ}$ (resp., $\overrightarrow{I_{j+1}I_j}$) by rotation of angle θ. We denote (R, S, T, U) the vertices of parallelogram $p(\theta)$ (see Figure 4).

Proposition 3 *The equations of lines (RU) and (ST) are given in $\mathcal{R}_0$ by $cs(\theta)Y - cs(\theta)X + cs(\theta) = 0$ and the equations of lines (UT) and (RS) are given in $\mathcal{R}_0$ by*

$aY - bX + cs(\theta) = 0$, where a, b are constants and $cs(\theta)$ is an expression of type $\alpha cos(\theta) + \beta sin(\theta) + \gamma$.

Proof : We first compute the equation of line (RU). Let $\mathcal{R}'$ be a reference frame parallel to $\mathcal{R}_0$ with origin at E_k and let us compute the equation of line (RU) in $\mathcal{R}'$. Let χ_0 be the orientation of vector $\overrightarrow{I_{j+1}I_j}$ when θ=0; the orientation of vector $\vec{i_\theta}$ is given by $\chi_0 + \theta$ and $\vec{i_\theta}$ is parallel to line (RU). The (signed) distance d_{RU} between (RU) and E_k satisfies $d_{RU} = \|\vec{u_\theta}\| \sin\alpha = \|\vec{u}\| \sin\alpha$ where $\alpha = (\vec{u_\theta}, \vec{i_\theta})$ is fixed (see Figure 4). Thus the equation of line (RU) in $\mathcal{R}'$ is given by :

$$(RU): \quad \cos(\chi_0 + \theta)Y - \sin(\chi_0 + \theta)X + \|\vec{u}\| \sin\alpha = 0$$

which gives the result in $\mathcal{R}_0$.

Let us compute now the equation of line (UT) in $\mathcal{R}'$. Let $\vec{v_\theta} = \vec{i_\theta} + \vec{u_\theta}$ and let d_{UT} be the (signed) distance between (UT) and E_k. We have :

$$d_{UT} = \|\vec{v_\theta}\| \cos\beta$$

$$d_{UT} = \|\vec{u} + \vec{i}\| \cos\beta$$

$$d_{UT} = \|\vec{u} + \vec{i}\| \cos(\psi_0 + \theta)$$

where β is angle $(\vec{n}, \vec{v_\theta})$ ($\vec{n}$ is a vector normal to edge e_k) and ψ_0 is angle β for θ=0. Line (UT) is parallel to edge e_k. Let (a, b) be a vector parallel to this direction. We have in $\mathcal{R}'$:

$$(UT) \; : \quad aY - bX + \|\vec{u} + \vec{i}\| \cos(\psi_0 + \theta) = 0$$

which gives the result in $\mathcal{R}_0$.

Similar results can be obtained for the lines (SR) and (TS). □

Parallelogram $p(\theta)$ intersects f along a segment denoted $]L(\theta), M(\theta)[$ and we have the following equalities :

$$\lambda(\theta) = \frac{1}{x_B - x_A} x_{L(\theta)} - \frac{x_A}{x_B - x_A}$$

$$\mu(\theta) = \frac{1}{x_B - x_A} x_{M(\theta)} - \frac{x_A}{x_B - x_A}$$

where x_P denotes the abscissa, in $\mathcal{R}_0$, of point P.

In order to find $\lambda(\theta)$ and $\mu(\theta)$, we have to compute $x_{L(\theta)}$ and $x_{M(\theta)}$.

Let us first look at the case where $x_{L(\theta)} = x_A$ which means that A lies in $p(\theta)$. As $p(\theta)$ is convex, A belongs to $p(\theta)$ iff A belongs to the four half planes defining $p(\theta)$. Thus, from Proposition 3, A belongs to $p(\theta)$ if θ satisfies four inequalities of type $cs(\theta) \geq 0$.

The set of θ satisfying such an inequality consists of one interval of $[0, 2\pi[$[1]; we conclude that $\lambda(\theta)$ equals 0 on a finite set of disjoint sub-intervals of $[0, 2\pi[$. For

[1] Intervals are considered on the unit circle

similar reasons, $\mu(\theta)$ equals 1 on a finite set of disjoint sub-intervals of $[0, 2\pi[$. All theses sub-intervals can be computed in constant time.

Suppose now that $x_{L(\theta)} \neq x_A$ (resp., $x_{M(\theta)} \neq x_B$). In that case, $L(\theta)$ (resp. $M(\theta)$) is the intersection between edge f and an edge a of $p(\theta)$. Two edges f and a intersect iff the endpoints of f are not on the same side of the line containing a and, simultaneously, the endpoints of a are not on the same side of the line containing f. Given two points Z_1 and Z_2 and a line L with equation $g(X, Y)$, Z_1 and Z_2 are not on the same side of L iff the product $g(X_{Z_1}, Y_{Z_1})g(X_{Z_2}, Y_{Z_2})$ is negative. The coordinates of the end points of a are of type $cs(\theta)$ thus, from the result above and Proposition 3, we deduce that f intersects a iff θ satisfies two inequalities of type $cs(\theta)cs(\theta) < 0$. As before, the set of θ where f and a intersect consist of a finite set of sub-intervals of $[0, 2\pi[$.

Because of the type of the equations of the lines supporting f and a (Proposition 1) it is easy to see that the abscissa of the intersection point between f and a is a function $F_a(\theta)$ of type $cs(\theta)$ when a is $[U, T]$ or $[S, R]$ and of type $cs(\theta)/cs(\theta)$ when a is $[R, U]$ or $[T, S]$.

Let $F_{min}(\theta)$ be the minimum of $\{F_A(\theta), F_{[R,U]}(\theta), F_{[U,T]}(\theta), F_{[T,S]}(\theta), F_{[S,R]}(\theta)\}$ and $F_{max}(\theta)$ be the maximum of $\{F_B(\theta), F_{[R,U]}(\theta)\ , F_{[U,T]}(\theta), F_{[T,S]}(\theta), F_{[S,R]}(\theta)\}$ where $F_A(\theta) = x_A$ (resp., $F_B(\theta) = x_B$) is defined on the interval(s) where $x_{L(\theta)} = x_A$ (resp., $x_{M(\theta)} = x_B$).

As $\lambda(\theta)$ is smaller than $\mu(\theta)$ we have :

$$\lambda(\theta) = \frac{1}{x_B - x_A} F_{min}(\theta) - \frac{x_A}{x_B - x_A}$$

$$\mu(\theta) = \frac{1}{x_B - x_A} F_{max}(\theta) - \frac{x_A}{x_B - x_A}$$

Because $[R, U], [U, T], [T, S]$ and $[S, R]$ make a closed curve and the functions $F_A(\theta)$, $F_B(\theta)$, $F_{[R,U]}(\theta)$, $F_{[U,T]}(\theta)$, $F_{[T,S]}(\theta)$, $F_{[S,R]}(\theta)$ are defined on a finite set of sub-intervals of $[0, 2\pi[$, functions $\lambda(\theta)$ and $\mu(\theta)$ are defined on a finite set of sub-intervals of $[0, 2\pi[$. It is clear from the above discussion that on each sub-interval of definition, $\lambda(\theta)$ (resp., $\mu(\theta)$) consists of a finite set of pieces, each of them being equal to the constant function 0 (resp., the constant function 1), a function of type $cs(\theta)$ or a function of type $cs(\theta)/cs(\theta)$.

We sum up these results in the following proposition :

Proposition 4 *Let $\lambda(\theta)$ and $\mu(\theta)$ be the functions associated to parallelogram $p(\theta)$ and edge f:*

1. *$\lambda(\theta)$ and $\mu(\theta)$ are defined on the same finite set of sub-intervals of $[0, 2\pi[$*
2. *On each of these sub-intervals, $\lambda(\theta)$ (resp., $\mu(\theta)$) is of one of the following types:*
   (a) 0 (resp., 1)
   (b) $cs(\theta)$

(c) $cs(\theta)/cs(\theta)$

3. *$\lambda(\theta)$ and $\mu(\theta)$ can be computed in constant time.*

3.2.2 Computation of $R(C)$

For each parallelogram $p_{jk}(\theta)$ ($j \in \{1,\cdots,m\}, k \in \{1,\cdots,n\}$) we can compute the associated functions $\lambda_{jk}(\theta)$ and $\mu_{jk}(\theta)$. Let us consider the graph of these functions. We call it the *diagram* associated to C (see Figure 5). To each pair $\lambda_{jk}(\theta), \mu_{jk}(\theta)$, we associate the region ρ_{jk} of $[0, 2\pi[\times[0,1]$ lying between $\lambda_{jk}(\theta)$ and $\mu_{jk}(\theta)$ (remember that $\lambda_{jk}(\theta)$ and $\mu_{jk}(\theta)$ are defined on the same sub-interval of $[0, 2\pi[$).

We define the $\Phi - region$ associated to contact C, denoted $\Phi(C)$ as $\Phi(C) = [0, 2\pi[\times[0,1]\backslash\bigcup_{j,k}\rho_{jk}$. To each free placement $(\theta, \overrightarrow{OM})$ of $R(C)$ corresponds a unique element (θ, ξ) of $\Phi(C)$ such that $\overrightarrow{OM} = \overrightarrow{OA} + \xi\overrightarrow{\tilde{A}\tilde{B}}$. This defines the one-to-one correspondance $\mathcal{F}$:

$$(\theta, \xi) \overset{\mathcal{F}}{\longleftrightarrow} (\theta, \overrightarrow{OA} + \xi\overrightarrow{\tilde{A}\tilde{B}})$$

In order to compute $R(C)$ we will first compute $\Phi(C)$ and then apply $\mathcal{F}$ to $\Phi(C)$. It is easy to see that $\Phi(C)$ can be decomposed into a finite set of non intersecting regions, which are limited by some functions $\lambda_{jk}(\theta)$ and $\mu_{j'k'}(\theta)$. Let ϕ be one of these regions and let $[\alpha, \beta]$ be the range of θ in ϕ.

Consider now the finite ordered set $\{\alpha_0 = \alpha, \cdots, \alpha_l = \beta\}$ of orientations corresponding either to an end point of a sub-interval where the functions $\lambda_{jk}(\theta)$ and $\mu_{j'k'}(\theta)$ defining ϕ keep a constant analytic form, or to the orientation of an intersection point between two such functions.

Let ϕ_i be the sub-region defined by $\phi_i = \{(\theta, \xi) \in \phi : \theta \in [\alpha_i, \alpha_{i+1}]\}$. We call it a *face* of the diagram (see Figure 6). The interiors of those sub-regions do not intersect and their union is exactly ϕ. Let l be the number of the sub-regions. To each ϕ_i, with $i \in \{1,\cdots,l-1\}$ corresponds two functions, say $\lambda_{jk}(\theta)$ and $\mu_{j'k'}(\theta)$ such that :

$$\phi_i = \bigcup_{\theta\in[\alpha_i,\alpha_{i+1}]} [(\theta, \lambda_{jk}(\theta)), (\theta, \mu_{j'k'}(\theta))] \tag{3}$$

Notice that ϕ_i is bounded by the two curves $\lambda_{jk}(\theta)$ and $\mu_{j'k'}(\theta)$ (which keep a constant analytic form) and possibly one line segment contained in the plane $\theta = \alpha_i$ and one line segment contained in the plane $\theta = \alpha_{i+1}$.

Proposition 5 *$\Phi(C)$ consists of a finite (possibily empty) set of regions which can be computed in time $O((mn + t)\log mn)$ where t is the number of triple-contacts involving contact C. In the worst case, $t = O(m^2n^2)$.*

Proof : We have to compute the different sub-regions ϕ_i. This can be done by a plane sweep algorithm quite similar to the one described in [11]. This latter finds the contour of the set theoretic difference (or of any other boolean operation) between

two families of polygons with a total number of N edges intersecting in t points in time $O((N+t)\log N)$.

The event point schedule [12] is initialized by the abscissae (orientations θ) of the end points of the sub-intervals $\lambda(\theta)$ and $\mu(\theta)$ are defined. This sequence of abscissae is dynamically updated during the execution of the sweep by inserting in the event point schedule the orientations of the intersections between functions $\lambda(\theta)$ and $\mu(\theta)$. The sweep line status consists of the list of functions $\lambda(\theta)$, $\mu(\theta)$ that intersect the sweep-line at orientation θ, sorted by increasing ordinates.

Instead of working with line segments as in [11], we must deal with pieces of monotone quadratic curves. Two points must be outlined. First, two curves may intersect in two points; thus, when the sweep line stops at a point belonging to two curves, say γ_1 and γ_2, we may have to compute the additional intersection $\gamma_1 \cap \gamma_2$. The second point is that we are able to compute intersections between two such curves. Indeed, due to Proposition 4, this problem reduces to the computation of the intersection between a line and a circle or to the intersection between a circle and a general conic. Both computations can be done in constant time [5].

This implies that the complexity of our algorithm is the same as the complexity of the one described in [11]. Here $N = O(mn)$ and t is the number of intersections between functions $\lambda(\theta)$ and $\mu(\theta)$. Notice that functions $\lambda(\theta)$ and $\mu(\theta)$ correspond to double-contacts (one of them being C). Thus an intersection between two such functions corresponds to a triple-contact (one of them being C). Thus the number t of intersections between curves $\lambda(\theta)$ and $\mu(\theta)$ is equal to the number of possible triple-contacts involving contact C. This achieve the proof. □

We are now in a position to compute $R(C)$. Contact-region $R(C)$ is the union of the images $\mathcal{F}(\phi_j)$ of all the sub-regions ϕ_j. Let ϕ_i be one of the sub-regions ϕ_j, bounded by two functions $\lambda_{jk}(\theta)$ and $\mu_{j'k'}(\theta)$.

By construction, both functions $\lambda_{jk}(\theta)$ and $\mu_{j'k'}(\theta)$ keep a constant analytic form on $[\alpha_i, \alpha_{i+1}]$. To segment $[(\theta, \lambda_{jk}(\theta)), (\theta, \mu_{j'k'}(\theta))]$ is associated by $\mathcal{F}$ a line segment, say $[P_i(\theta), Q_i(\theta)]$ and, from Equation (3) we have :

$$\mathcal{F}(\phi_i) = \bigcup_{\theta \in [\alpha_i, \alpha_{i+1}]} [P_i(\theta), Q_i(\theta)]$$

More precisely, using correspondence $\mathcal{F}$, we have (AB is the edge involved in contact C) :

$$\overrightarrow{OP_i(\theta)} = \overrightarrow{O\tilde{A}} + \lambda_{jk}(\theta)\overrightarrow{\tilde{A}\tilde{B}}$$

and

$$\overrightarrow{OQ_i(\theta)} = \overrightarrow{O\tilde{A}} + \mu_{j'k'}(\theta)\overrightarrow{\tilde{A}\tilde{B}}$$

Vectors $\overrightarrow{OP_i(\theta)}$ and $\overrightarrow{OQ_i(\theta)}$ have the generic form $\overrightarrow{O\tilde{A}} + \xi(\theta)\overrightarrow{\tilde{A}\tilde{B}}$ where $\xi(\theta)$ keeps a constant analytic form on $[\alpha_i, \alpha_{i+1}]$. Remember that $\overrightarrow{\tilde{A}\tilde{B}} = T_{\overrightarrow{J_\theta O}}(\overrightarrow{AB})$, in case of

a vertex-edge contact, and, $\overrightarrow{\tilde{A}\tilde{B}} = T_{\overrightarrow{JO}}(\overrightarrow{AB_\theta})^*$ in case of an edge-vertex contact (see Section 3.2). As J has been taken as the origin, in case of a vertex-edge contact, we have : $\overrightarrow{\tilde{A}\tilde{B}} = \overrightarrow{AB}$ and $\overrightarrow{O\tilde{A}} = \overrightarrow{OA} + R_\theta(\overrightarrow{JO})$. Thus :

$$\overrightarrow{O\tilde{A}} + \xi(\theta)\overrightarrow{\tilde{A}\tilde{B}} = \overrightarrow{OA} + R_\theta(\overrightarrow{JO}) + \xi(\theta)\overrightarrow{AB}$$

In case of an edge-vertex contact, we have : $\overrightarrow{\tilde{A}\tilde{B}} = R_\theta(\overrightarrow{BA})$ and $\overrightarrow{O\tilde{A}} = \overrightarrow{A_\theta O} + \overrightarrow{OJ} = \overrightarrow{A_\theta J}$. Thus :

$$\overrightarrow{O\tilde{A}} + \xi(\theta)\overrightarrow{\tilde{A}\tilde{B}} = \overrightarrow{A_\theta J} + \xi(\theta)R_\theta(\overrightarrow{BA})$$

This proves that the coordinates of $P_i(\theta)$ and $Q_i(\theta)$ keep a constant analytic form on $[\alpha_i, \alpha_{i+1}]$. Thus $\mathcal{F}(\phi_i)$ is a ruled surface which can be computed in constant time from ϕ_i. We call it a *face* of $R(C)$. We conclude this section by the following proposition :

Proposition 6 *$R(C)$ consists of a finite (possibily empty) set of faces which can be computed in time $O((mn + t)\log mn)$ where t is the number of triple-contacts involving contact C. In the worst case, $t = O(m^2n^2)$.*

3.2.3 Faces, edges and vertices of $R(C)$

As we have seen in the previous section, contact-region $R(C)$ is a finite set of faces $\mathcal{F}(\phi_i)$ $(i = 1, ..., k)$ where ϕ_i is a sub-region of $\Phi(C)$.

Such a face is bounded by the two curves $\mathcal{F}(\theta, \lambda_{jk}(\theta))$ and $\mathcal{F}(\theta, \mu_{j'k'}(\theta))$ (which keep a constant analytic form) and possibly one line segment contained in the plane $\theta = \alpha_i$ and one line segment contained in the plane $\theta = \alpha_{i+1}$. Theses curves are called the *edges* of the face.

We distinguish two types of edges (see Figure 7). The edges of *type 1* belong to two different faces of $R(C)$. The edges of *type 2* belong to only on face of $R(C)$. Each edge of type 2 lies on the boundary of $R(C)$ and thus is a set of double-placements involving a pair of contacts, one of them being C. The edges of type 1 are always straight; the edges of type 2 may be either straight or curved.

An intersection between two edges of a face is called a *vertex*. A vertex which is the intersection of two edges of type 2 is a free triple-contact. Any edge contains at least one vertex which is a free triple-contact. Hence we have :

Proposition 7 *The number of faces, edges and vertices are linearly related to the number of free triple-contacts.*

4 Computation of $P(I, E)$

4.1 The case where only translations are allowed

From Proposition 1, the boundary of $P(I, E)$ is the union of the $R(C)$for all the mn contacts C. We have shown in Section 3.1 that each $R(C)$ is a collection of $O(mn)$

disjoint line segments which can be computed in $O(mn \log mn)$ time. The end-points of these segments are free double-contacts . Moreover any free double-contact (C, C') is an end-point of one segment S of $R(C)$ and of one segment S' of $R(C')$. Hence the two segments S and S' are adjacent edges in $\partial P(I, E)$ and $\partial P(I, E)$ is a set of closed polygonal curves with a total of $O(m^2n^2)$ edges. If we label the end-points of the segments of $R(C)$ with the corresponding double-contacts, we obtain in $O(m^2n^2)$ time the adjacency relationships between the edges of $\partial P(I, E)$. Lastly, we observe that the set of free placement $P(I, E)$ can be trivially deduced from its boundary. Thus we have :

Proposition 8 *$P(I, E)$ can be computed in time $O(m^2n^2 \log mn)$.*

4.2 The general case

In order to compute a complete description of the boundary of $P(I, E)$, we compute first the set of all its faces by computing $R(C)$ for all the mn contacts C. As stated by Proposition 6, computing one contact-region $R(C)$ takes $O((mn+t) \log mn)$ time, where t is the number of triple-contacts involving C. Thus computing all the faces of $\partial P(I, E)$ takes $O((m^2n^2 + T) \log mn)$ time, where T is the total number of (not necessarily free) triple-contacts. In the worst case, $T = \Omega(m^3n^3)$ as shown in the appendix.

In addition, we compute the adjacency relationships between the faces of $\partial P(I, E)$ as follows. Two faces of $\partial P(I, E)$ are adjacent by an edge or a portion of an edge. This yields two types of adjacency relationships, depending on the type of the edge shared by the two faces (see Section 3.2.3).

Recall that $R(C)$ has been computed by a sweep-plane algorithm. The adjacency relationships of type 1 can be easily computed during the sweeps computing the contact-regions. At each stop of the sweep-line computing a contact-region $R(C)$, we output new adjacency relationships of type 1 between faces of $R(C)$. This can be done without increasing the overall complexity of the sweep-plane algorithm.

Let us consider now the adjacency relationships of type 2. We label each edge of type 2 by the corresponding double-contact. While computing $R(C)$, we associate to each double-contact D the list, sorted by increasing (θ, ξ), of the faces which have an edge of type 2 labelled by D. Notice that D appears in exactly two contact-regions, $R(C)$ and $R(C')$. Thus to each double-contact D are associated two sorted lists of faces, denoted $L(C)$ and $L(C')$ (see Figure 8). A face of $L(C)$ is adjacent to a face of $L(C')$ iff there exists a θ (a ξ if θ is constant along D) for which they are both defined. Then, by merging in time $O(|L(C)| + |L(C')|)$ the two lists $L(C)$ and $L(C')$, we obtain the adjacency relationships of type 2 along D. As a face has at most four edges of type 2, computing the adjacency relationships of type 2 requires time proportional to the number of faces which is no more than $O(T)$ by Proposition 7. Lastly, we observe that the set of free placement $P(I, E)$ can be trivially deduced from its boundary. Thus we have :

Proposition 9 *$P(I, E)$ can be computed in time $O((m^2n^2 + T) \log mn)$ where T*

is the total number of (not necessarily free) triple-contacts. In the worst-case, $T = \Omega(m^3n^3)$.

5 Final remarks

5.1 Relative positions of I and E

The set $P(I, E)$ is the closure of the set of placements of I such that the boundaries of I and E do not intersect. But we cannot immediatly deduce from $P(I, E)$ what are the relative positions of I and E. When I and E are connected, there are three different situations : $I \subset E, E \subset I$ or $I \bigcap E = \emptyset$. When I and-or E are not connected, these situations occur for the differents connected components of I and-or E.

When I is connected (E may be not connected), each connected component of $P(I, E)$ is a set of placements that either fit I inside E, fit a connected component of E inside I or fit I outside E. The distinction between the three situations can be done without increasing the complexity of the algorithm. Indeed, it is sufficient to perform the test for one point of a face bounding this component. Such a face is a face of a contact-region for some free contact $C = (J, f)$. For a placement on this face, the situation depends only on the type of J (convex or non convex) and on the local relative positions of I and E around J (see Figure 9). The distinction between the different cases can clearly be done in constant time. When I is not connected, each connected component I_j of I can be in one of the three above situations. Each connected component of $P(I, E)$ is a set of placements where the situations for the different I_j are fixed. For the connected components of $P(I, E)$ whose boundary contains a face corresponding to a contact involving a vertex of I_j, the distinction between the three situations can be done in constant time as described above. For the connected components of $P(I, E)$ where no vertex of I_j is involved, we need, in addition, to check for one point of I_j whether it belongs to E or not.

5.2 Actual time complexity

The time complexity of our algorithm has been shown to be $O((m^2n^2 + T) \log mn)$ where T is the number of triple contacts which may be $\Omega(m^3n^3)$ as shown in the appendix. Thus our algorithm is close to optimal in the worst-case. Moreover, its actual complexity in practice may be much smaller. To enlight this point, we consider two situations : the case where I is a segment and the case where the complexity of E is locally bounded.

In case that I is a line segment, $T = O(n^2)$. Indeed, a triple-contact corresponds to one of the three possible cases :

1. An endpoint of I is in contact with a vertex of E and the other end point of I is in contact with an edge of E.

2. An endpoint of I is in contact with a vertex of E and a vertex of E is in contact with the interior of I.

3. The interior of I is in contact with two vertices of E and an endpoint of I is in contact with an edge of E.

4. Three vertices of E are in contact with the interior of an edge of I.

It is clear that the three first cases can only occur $O(n^2)$ times. If no set of $O(n^{2/3})$ vertices of E are colinear the fourth case can also only occur $O(n^2)$ time. *The complexity of our algorithm is thus $O(n^2 \log n)$ when I is a line-segment.*

Let us consider now the following typical situation often encounted in practice (referred to as a situation of *bounded local complexity* in [14]). The number m of vertices of I is small (and thus can be considered to be a constant) and, in addition, the edges of E are not concentrated near each other compared with the size of I (its diameter l). More precisely, let us define as in [14] the l-neighbors of a vertex e of E as the edges and vertices of E which meet the closed disc of radius l about e. Let us suppose that for each vertex of E the number of its l-neighbors is bounded by a fixed number r independant of n. It is clear that if two edges e_1 and e_2 are involved in a double-contact, e_1 is an l-neighbor of an endpoint of e_2 or e_2 is an l-neighbor of an endpoint of e_1. Let us call such a pair (e_1, e_2) a *valid* pair. As shown by Sifrony and Sharir [14], all such valid pairs can be found in time $O(n \log n)$. We store, for each edge e of E the list of edges e' such that (e, e') is valid (such a list has less than r elements).

Consider now the computation of a contact region $R(C)$ where C is a vertex-edge contact involving edge e of E (in case of an edge-vertex contact, the two edges of E sharing the contact vertex are to be considered). The parallelograms we need to consider for the computation of $R(C)$ are necessarily those corresponding to an edge of I and an edge e' of E such that the pair (e, e') is valid. There are $O(1)$ such parallelograms which can be computed in time $O(1)$. This shows that the computation of $R(C)$ takes $O(1)$ time and thus, the computation of $\partial P(I, E)$ takes $O(n)$ time.

In conclusion, *for situations of bounded local complexity, our algorithm can be easily adapted to run in $O(n \log n)$ time.*

5.3 Polyhedron placement under translation

Our algorithm can be easily extended to solve the analogous 3-dimensional problem when only translations are allowed. Let I and E be two polyhedra with respectively m and n triangular faces. A parallelogram p_{ij} of Section 2 is now replaced by a convex polyhedron P_{ij} with at most nine vertices, namely the Minkowski difference between two triangles of 3-space –which can be computed in constant time. Each contact-region is the set theoretic difference between a triangle and the union of P_{ij}, i.e. the union of the Minkowski differences between all pairs of faces of I and E. Each contact-region $R(C)$ can be computed by the sweep-plane technique of [11] in

$O((mn+t)\log mn)$ time where t is the number of (not necessarily free) triple-contacts involving contact C. As there are mn such contact-regions, we can compute $P(I,E)$ for two polyhedra in time $O((m^2n^2+T)\log mn)$, where T is the total number of (not necessarily free) triple-contacts. In the worst-case, $T = O(m^3n^3)$. Thus our algorithm is close to optimal in the worst-case since the solution may consist of $\Omega(m^3n^3)$ faces.

6 Conclusion

We have presented a general and simple algorithm which computes all placements for a polygonal object I (with m edges) which is free to translate and/or to rotate but not to intersect another polygonal object E. The worst-case time complexity of our algorithm is $O(m^3n^3\log mn)$ which is close to optimal. In the case where I is a line segment, the complexity is reduced to $O(n^2\log n)$ and for situations of bounded local complexity the complexity is reduced to $O(n\log n)$. The algorithm has been implemented. Figure 10 shows an example; the two polygons are shown in (a), a diagram is shown in (b) and shaded graphics displays of the subset of $P(I,E)$ corresponding to placements with $I \subset E$ are shown in (c-e). The θ-axis is vertical and, for clarity, θ ranges between 0 and 4π.

The description of the boundary of free space computed by our method is well suited for motion planning. Indeed, each face has a simple analytical form which allows to easily compute a path between two points of a given face. Moreover, we have computed the topological structure of the set of faces. Actually, it is possible to compute a free motion between any two placements in time proportional to the number of faces. Details can be found in [2].

References

[1] AVNAIM F., BOISSONNAT J.D., Simultaneous containment of several polygons, 3rd ACM Symp. on Computational Geometry, Waterloo (June 1987).

[2] AVNAIM F., BOISSONNAT J.D., B. FAVERJON, A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles, IEEE Conf. on Robotics and Automation, Philadelphia (1988).

[3] ALBANO A., SAPUPPO G., Optimal Allocation of Two-Dimensional Irregular Shapes Using Heuristic Search Methods, IEEE Trans. on Systems, Man and Cybern., Vol. SMC-10, No 5, May 1980.

[4] BAKER B.S., FORTUNE S.J., MAHANEY S.R., Inspection by Polygon Containment, 22th Allerton Annual Conf. on Communications, Control and Computing, 1984, pp. 91-100.

[5] BERGER M., Géométrie Volume 4: Formes quadratiques, coniques et quadriques, CEDIC/Fernand Nathan (1978).

[6] CHAZELLE B., The polygon containment problem, in Advances in computer research, Vol. 1 (F.P. Preparata, ed.), JA Press, pp. 1-32.

[7] FORTUNE S.J., Fast algorithms for polygon containment, Automata, Languages and Programming, in Lecture Notes in Computer Science 194,Springer Verlag, pp. 189-198

[8] GUIBAS L., RAMSHAW L., STOLFI G., A kinematic framework for computational geometry, Proc. IEEE Symp. on Foudations of Comput. Sci. (1983), pp. 74-123.

[9] KEDEM K., SHARIR M., An efficient motion planning algorithm for a convex polygonal object in 2-dimensional polygonal space, Tech. Rept. No 253, Comp. Sci. Dept., Courant Institute, (Oct. 1986).

[10] LEVEN D., SHARIR M., On the number of critical free contacts of a convex polygonal object moving in 2-D polygonal space, Discrete and Computational Geometry, Vol. 2, No 3, 1987.

[11] OTTMAN T., WIDMAYER P., WOOD D., A fast algorithm for Boolean mask operations, Computer Vision, Graphics and Image Processing, Vol. 30, 1985, pp. 249-268.

[12] PREPARATA F.P., SHAMOS M.I., Computational geometry: an introduction, Springer Verlag, 1985.

[13] SCHWARTZ J.T., SHARIR M., On the piano mover's problem I. The case of a two dimensional rigid polygonal body moving amidst polygonal barriers, Comm. Pure Appl. Math., 36 (1983), pp. 345-398.

[14] SIFRONY S., SHARIR M., A New Efficient Motion Planning Algorithm for a Rod in Two-Dimensional Polygonal Space, Algorithmica (1987) 2, pp. 367-402.

Appendix: $\Omega(m^3n^3)$ is a lower bound for placement by translation and rotation

We present here an example where the number of free triple-contacts is $\Omega(m^3n^3)$ and consequently, the number of faces of $\partial P(I,E)$ is $\Omega(m^3n^3)$.

Let us consider a circle C with center Ω and radius R on which n points (the polygonal region E) are regularly distributed. The region I consists of three groups B_1, B_2, B_3 of $(m+3)/6$ parallel straight line segments with length $2R$. The distance between two consecutive segments of B_i $(i = 1,2,3)$ is ϵ.

These three groups are attached to a small equilateral triangle O and the angle between any two segments belonging to two consecutive groups is equal to $\frac{2\pi}{3}$ (see Figure 11).

Let us call L the distance between two consecutive points on C and let l be the maximum distance between the two extreme segments of a group B_i. We have $L = 2R\sin\frac{\pi}{n}$ and $l = \epsilon(m-3)/6$. For a sufficiently small ϵ, at most one point of E can lie between two segments of a given group B_i.

We claim now that the number of free triple-contacts is in this case $\Omega(m^3n^3)$. Indeed, let us call b a segment of B_1 and e a point of E such that b touches e and the center of O belongs to the line $(e\Omega)$. Let e' be the point opposite e on C. When O goes from e to e', the $\frac{m}{3}$ line segments of B_2 go through $\Omega(n)$ points of E, yielding $\Omega(mn)$ free double-contacts, all involving the contact (e,b). Choosing any other line segment in B_1 and any other point in E yields $\Omega(m^2n^2)$ free double-contacts involving B_1 and B_2.

Let us consider now a given free double-contact $(e_1,b_1),(e_2,b_2)$, involving B_1 and B_2. This double-contact is realized for any position of O in disc C such that the intersection point b_{12} between b_1 and b_2 sees the line segment $[e_1e_2]$ under a constant angle $\frac{2\pi}{3}$. The locus of such positions is a circle arc denoted a passing trough e_1 and e_2, contained in disc C. When b_{12} moves on a, the line segments of B_3 go through a circle arc of C whose length is at least $\frac{2\pi}{3}$, meeting in that movement $\Omega(n)$ points of E. Thus the number of free triple-contacts involving $(e_1,b_1),(e_2,b_2)$ is $\Omega(mn)$ and the total number of free triple-contacts is $\Omega(m^3n^3)$.

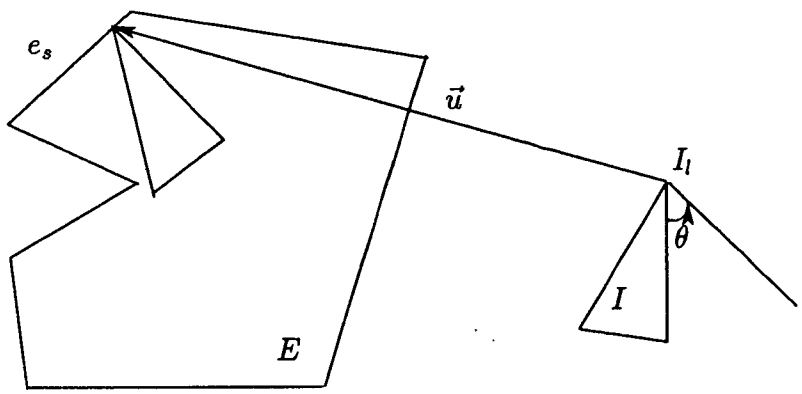

Figure 1: Example of a free contact (I_l, e_s)

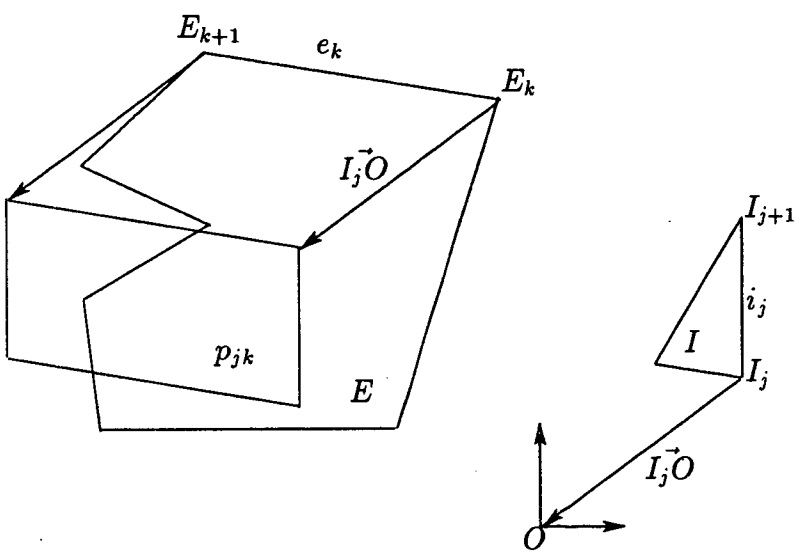

Figure 2: Parallelogram p_{jk}

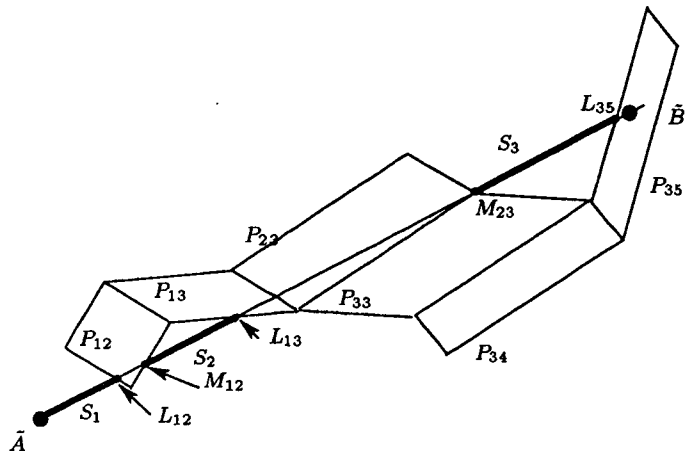

Figure 3: For the proof of Proposition 2

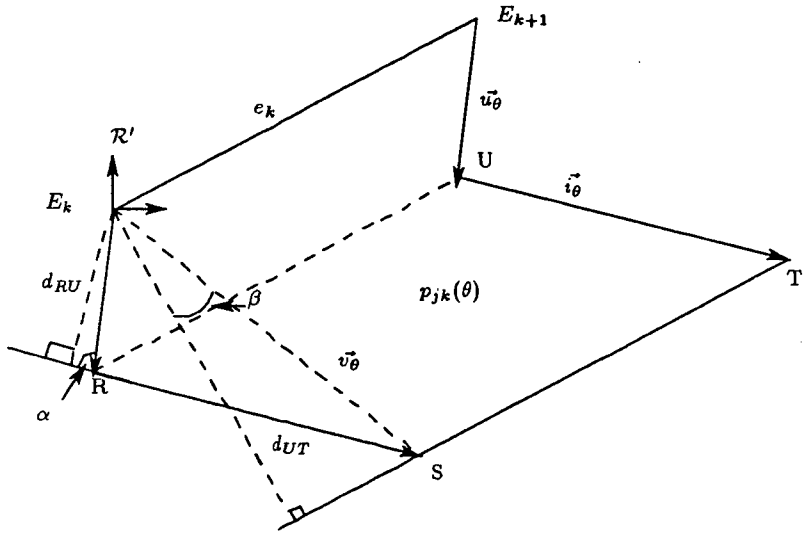

Figure 4: parallelogram $p_{jk}(\theta)$

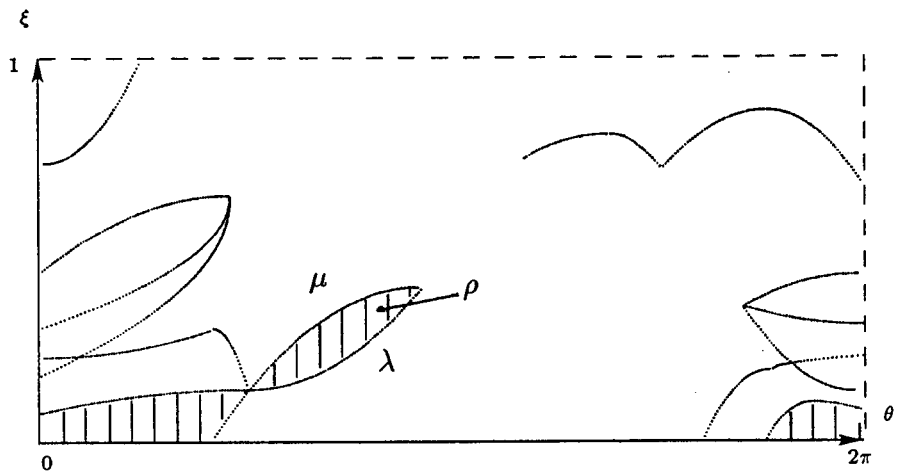

Figure 5: The diagram associated to C

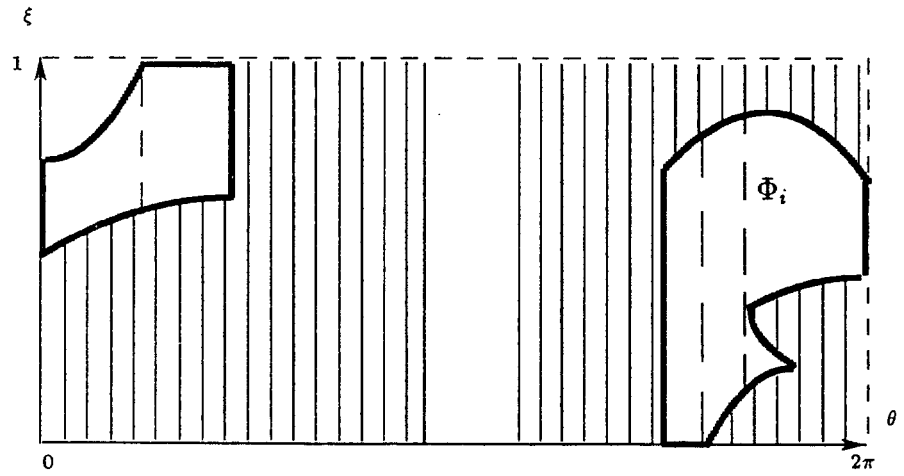

Figure 6: The sub-regions ϕ_i of ϕ

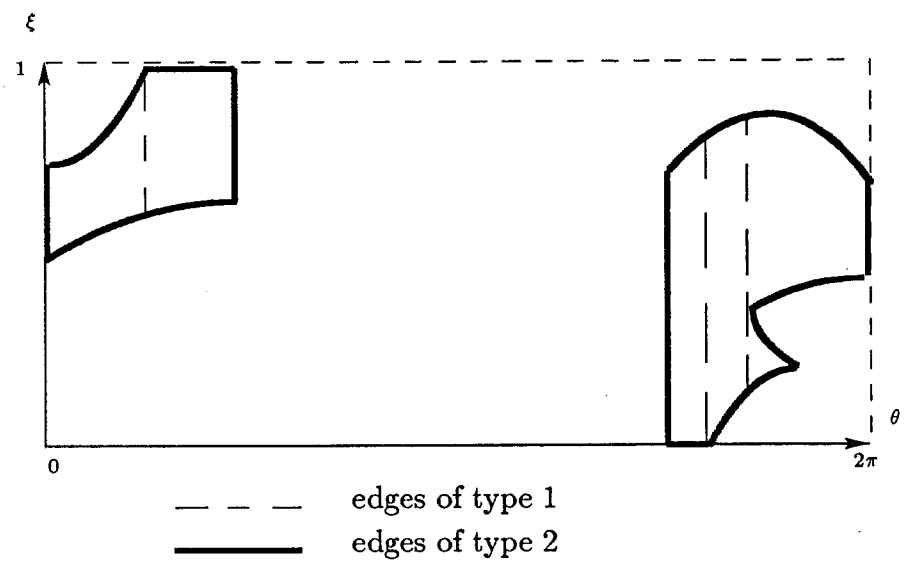

Figure 7: The two types of edges of ϕ

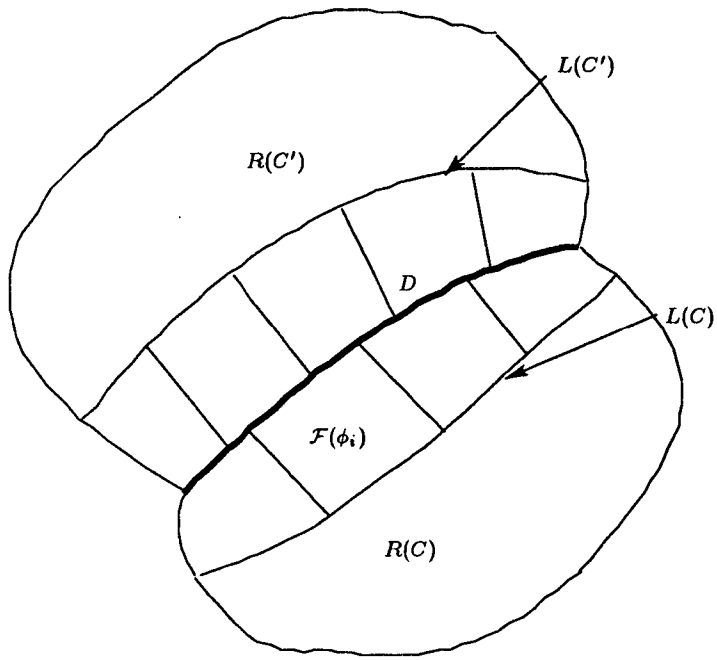

Figure 8: The two lists $L(C)$ and $L(C')$

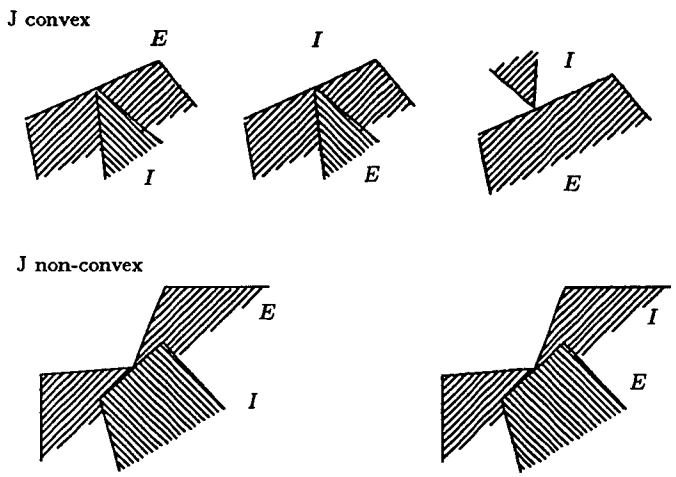

Figure 9: The local relative positions of I and E

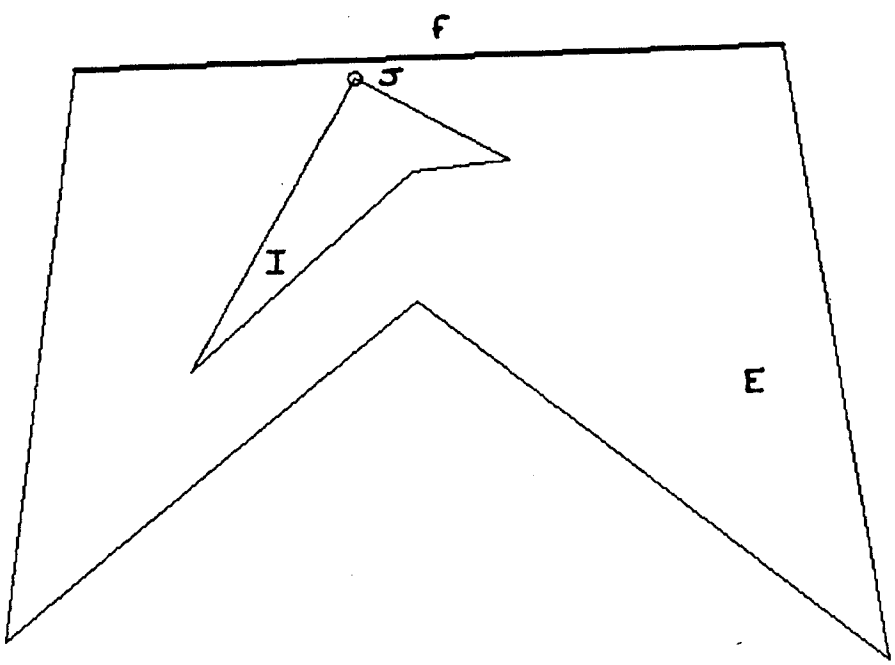

Figure 10.a: The two polygons I and E

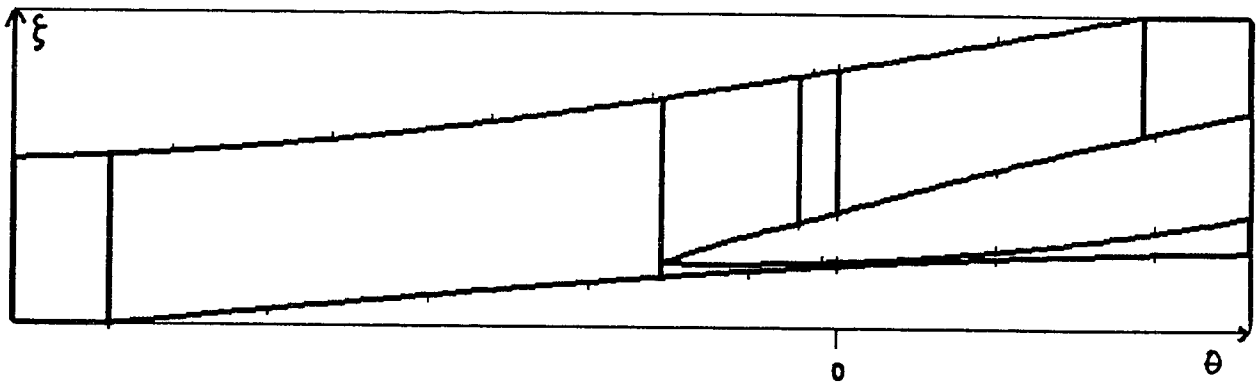

Figure 10.b: Diagram associated to the contact (J, f)

Figure 10.c: Shaded graphics display of $P(I, E)(I \subset E)$; perspective view

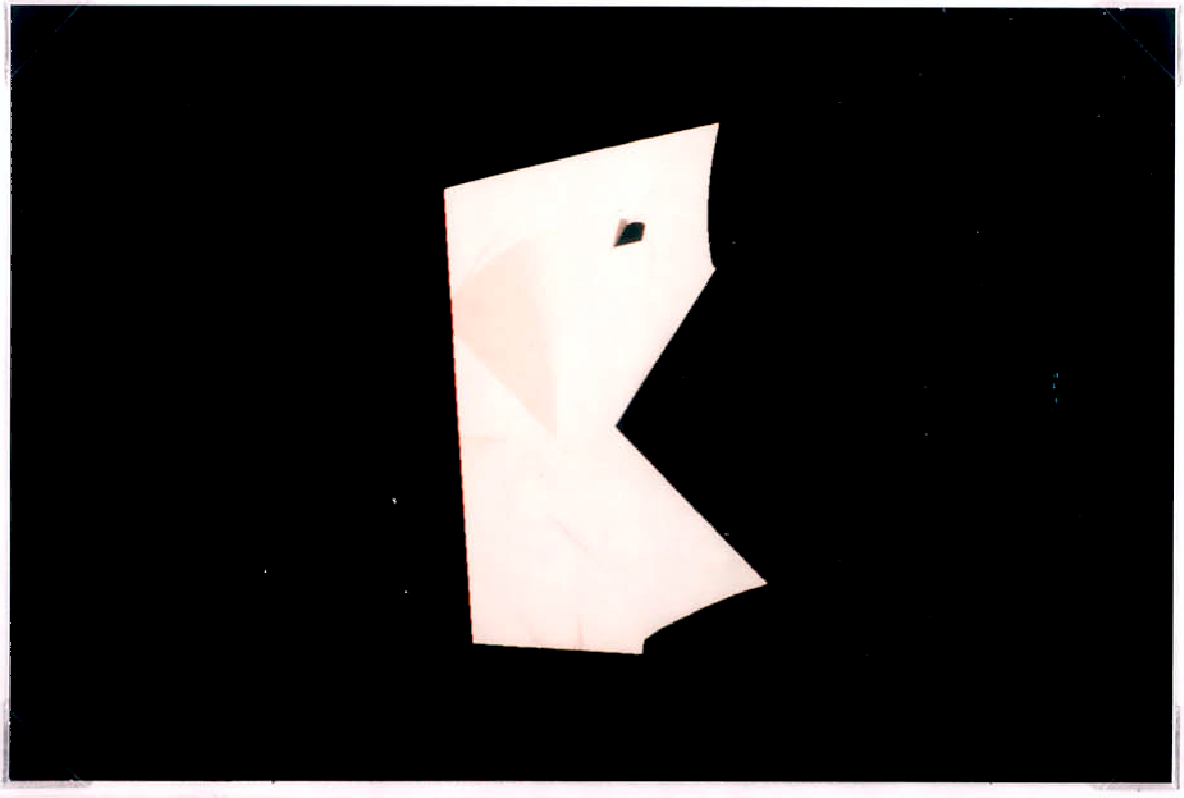

Figure 10.e: Shaded graphics display of $P(I, E)(I \subset E)$; top view

Figure 10.d: Shaded graphics display of $P(I, E)(I \subset E)$; front view

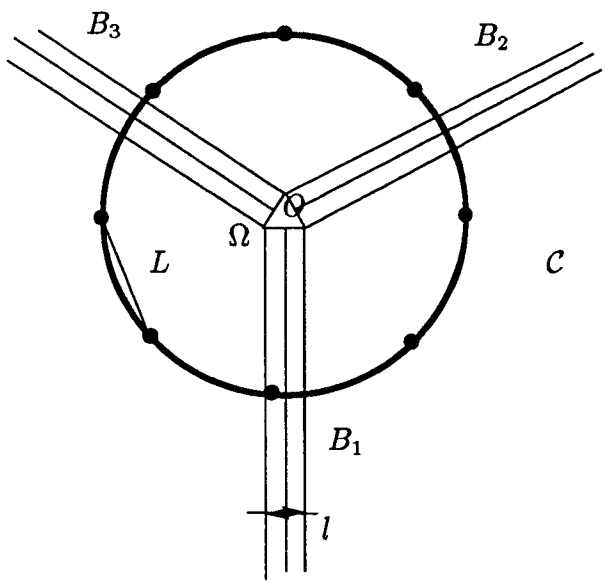

Figure 11: Lower bound $\Omega(m^3n^3)$

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique