2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Planning Optimal Independent Contact Regions for Two-Fingered Force-Closure Grasp of a Polygon

Thanathorn Phoka, Pawin Vongmasa, Chaichana Nilwatchararang,Peam Pipattanasomporn and Attawith Sudsang

Abstract— **This paper addresses the problem of optimizing the maximal independent contact region for two-fingered force-closure grasp of a rigid object in 2D. Existing methods for optimizing this criterion considered only independent graspable regions on a given pair of edges. We propose an algorithm that takes nearby edges into consideration so that larger independent contact regions can be obtained. Our method takes the input polygons, computes graspable regions for each pair of edges, merge all adjacent regions together, and then find the best independent contact region inscribed in those regions. Two different criteria to define the *best* independent contact region are studied. The first criterion maximizes area of the axis-parallel rectangle in the configuration space, while the other criterion maximizes the smaller side's length of the rectangle. For a given object with n vertices, the first criterion can be optimized using the algorithm from Karen Daniels *et al.* in $O(n^2 \log^2 n)$ time, while the other criterion can be optimized using the algorithm from Evanthia Papadopoulou and D. T. Lee in $O(n^2 \log n)$ time.**

I. INTRODUCTION

Grasping is an important operation in many manufacturing processes. Especially, two-fingered grasping modules are extensively applied in several procedures due to their simple and robust operation. To ensure that the object is grasped securely, the classical *force-closure* condition is considered. A grasp of an object achieves force-closure when it can resist any external wrench exerted on the grasped object. The well-known qualitative test for a force-closure grasp is to check whether the contact wrenches of the grasp positively span the whole wrench space [1]. This is equivalent to checking whether the convex hull of the primitive contact wrenches contains the origin [2].

Other approaches of qualitative test for a force-closure grasp by considering the workspace, not the wrench space, were also investigated. Nguyen [3] proposed a geometric method for testing two-finger force-closure grasps on polygonal objects. Ponce *et al.* proposed the concept of non-marginal equilibrium which implies the force-closure property. Based on this concept, the qualitative tests of three-finger grasps for polygonal objects [4] and four-finger grasps for polyhedral objects [5] were proposed. Alternatively, Blake considers both the workspace and the configuration space [6], and classifies planar grasps into three types using the symmetry set, the anti-symmetry set, and the critical set along with the friction function.

Quantitative tests of force-closure grasps are also considered to define the quality of grasps. Ferrari and Canny [7] considered the best performance in resisting external wrenches as the optimality criterion. Based on this criterion, Zhu and Wang [8] addressed the problem of planning optimal grasps that maximize the Q distance and expresses the best performance in firmly holding an object while resisting external wrench loads. Zhu *et al.* [9] solved the same problem by optimizing the pseudodistance function.

Most methods mentioned above are used to determine grasps that require precision of fingertip on the objects. To allow some positioning errors, the notion of *independent contact regions* was introduced by Nguyen [3]. In short, an independent contact region is a parallel-axis rectangular region in fingers' configuration space which represents areas on object's boundary where fingers can be placed independently to compose a force-closure grasp. In [3], Nguyen also showed how to geometrically determine independent contact regions for two-fingered grasps of a polygon. Tung and Kak [10] attacked the completeness of the previous work and proposed an algorithm which is correct and complete. Recently, Cornella and Suarez investigated an algorithm of determining independent grasp regions on 2D discrete objects [11]. A four frictionless grasp is considered. The algorithm determines the independent regions of two fingers when the locations of the other two fingers are given. Nancy S. Pollard points out that more-than-sufficient contact points can give more quality and flexibility of grasp, and in [12], a fast algorithm to synthesize many-contact grasps that preserve some properties of an example grasp is presented.

In order to find the *best* independent contact region, one needs to define what *best* means. There have been many different definitions of the best independent contact region due to different purposes and constraints of grasping devices. The two popular criteria are: (1) the largest n-cube, and (2) the largest rectangular region (product of lengths on every axis). Using the first criterion, the optimization can be done by linear programming as discussed in [4] and [5]. Faverjon and Ponce [13] tackled the problem of two-fingered grasping on curved objects using the second criterion. In their work, a numerical optimization algorithm was presented, but they could not guarantee the algorithm's completeness. Cornella and Suarez [14] presented an approach to determine

This research is financially supported in part by the Thailand Research Fund through the Royal Golden Jubilee Ph.D. program and the 90^{th} Anniversary of Chulalongkorn University Fund through the Ratchadapisek-somphot Fund, both are greatly appreciated.

T. Phoka, C. Nilwatchararang, P. Pipattanasomporn, and A. Sudsang are with the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, 10330, Thailand thanathorn.p@gmail.com, chaichana.n@gmail.com, omegale@gmail.com, attawith@gmail.com

P. Vongmasa is with the Institute for Computational and Mathematical Engineering, Stanford University, CA 94305, USA tunococ@stanford.edu

independent contact regions on polygonal objects considering arbitrary number of friction or frictionless contacts on given edges. Their approach subdivides configuration space so that the graspable region in each subdivision is convex, then computes the independent contact region in each subdivision.

So far, existing works in the literature on determining independent contact regions require that each contact point must lie on a given edge. This restriction hinders the discovery of independent contact regions across multiple adjacent edges. Therefore, in this paper, we present how independent contact regions across edges can be found for the case where there are two fingers with friction. Our method requires that the input object consists of non-self-intersecting simple polygons. The proposed algorithm for reporting optimal independent contact region of a polygonal object with n vertices runs in $O(n^2 \log n)$ under the n-cube optimizing criterion, and in $O(n^2 \log^2 n)$ under the largest rectangle optimizing criterion. Note that our method does not require that the object be simple or connected.

II. BACKGROUND

In two dimensions, a hard finger in contact with some object at a point $\mathbf{x} = (x_1, x_2)$ exerts a force $\mathbf{f} = (f_1, f_2)$ parallel to the normal vector of the contact surface. The force $\mathbf{f}$ will generate torque $\det(\mathbf{x}, \mathbf{f}) = x_1 f_2 - f_1 x_2$ with respect to the origin. The force and the torque are combined to form a three dimensional *wrench* $\mathbf{w} = (\mathbf{f}, \det(\mathbf{x}, \mathbf{f}))$. In the absence of friction, all forces that can be produced at a single contact point are normal to the contact surface and point inward. Therefore, the force that can be exerted is unique up to positive scaling. Since $\det(\mathbf{x}, c\mathbf{f}) = c \det(\mathbf{x}, \mathbf{f})$, the wrench is also unique up to positive scaling.

In the presence of friction, a single contact point can exert forces in different directions. The set C of all forces that can be exerted at the contact point is called the *friction cone*. As a result, the set of wrenches that can be produced is:

$$W = \{(\mathbf{f}, \det(\mathbf{x}, \mathbf{f})) : \mathbf{f} \in C\}.$$

A d-finger grasp is defined geometrically by the position $\mathbf{x}_i (i = 1, ..., d)$ of the fingers on the boundary of the grasped object. We can associate with each grasp the set of wrenches $W \subset \Re^3$ that can be exerted by the fingers. If we denote by W_i the wrench set associated with the i^{th} finger, we have

$$W = \left\{ \sum_{i=1}^{d} \mathbf{w}_i : \mathbf{w}_i \in W_i \text{ for } i = 1, ..., d \right\}.$$

Definition 1: A two dimensional grasp is said to achieve force-closure when the corresponding wrench set W is equal to $\Re^3$.

In other words, a grasp achieves force-closure when any external wrench can be balanced by wrenches at the fingertips. A somewhat weaker condition is equilibrium, defined below.

Definition 2: A grasp is said to achieve equilibrium when there exist forces (not all of them being zero) in the friction cones at the fingertips such that the sum of the corresponding wrenches is zero.

It is formally shown in [3] and [4] for two finger cases that a sufficient condition for force-closure is non-marginal equilibrium grasps, i.e., grasps such that the forces achieving equilibrium lie strictly inside the friction cones at the fingertips.

Proposition 1: A sufficient condition for two-fingered force-closure is non-marginal equilibrium

That is, grasps achieving equilibrium with non-zero forces for some friction coefficient achieve force-closure for any strictly greater friction coefficient. Due to [3], the following proposition characterizes two-finger equilibrium.

Proposition 2: A necessary and sufficient condition for two points to form an equilibrium grasp with non-zero contact forces is that the line joining both points lies completely in the two double-sided friction cones at the points.

III. REPRESENTING FORCE-CLOSURE GRASPS

Let us now state the problem. The object of interest does not have to be connected nor simple, but its boundary must not be self-intersecting. Its boundary can be broken into simple polygons. We are concerned with the problem with two simple polygons because if there are more than two simple polygons that define the object, we can pick two at a time and run the same algorithm over all possible pairs. (We are only interested in two-fingered grasps.)

The configuration of the problem consists of two parameters, each of which defines where a finger is placed along the boundary of the grasped object. In this section, we describe how to represent and construct the configuration space that characterizes all force-closure grasps.

We now define entities of a polygon needed in our consideration as follows. A simple polygon P is defined by n distinct vertices $\mathbf{v}_i \in \Re^2$ where $i \in \mathbf{Z}_n$ [1]. We will assume that $\mathbf{v}_i$ are arranged counterclockwise if it represents the outer boundary of the object, or arranged clockwise if it represents the hole inside the object. Edges E_i are line segments with endpoints $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$. Every point $\mathbf{p}$ on P's boundary can be mapped to the length of curve measured counterclockwise from $\mathbf{v}_0$ to $\mathbf{p}$ along the boundary. We will write $length(\mathbf{p})$ to represent such length. Lengths of E_i can be computed by the equation $l_i = \|\mathbf{v}_{i+1} - \mathbf{v}_i\|$. It is obvious that $L_i = length(\mathbf{v}_i) = \sum_{j \in \mathbf{Z}_i} l_j$. We denote by L the total length of P's boundary, which can be computed by $L = \sum_{i \in \mathbf{Z}_n} l_i$.

Next, let us define tangents of E_i as $\mathbf{t}_i = (\mathbf{v}_{i+1} - \mathbf{v}_i)/l_i$. The normal vectors $\mathbf{n}_i$ of E_i are unit vectors that are perpendicular to $\mathbf{t}_i$ and point inward ($\mathbf{n}_i$ can be obtained by rotating $\mathbf{t}_i$ $\pi/2$ radian counterclockwise). The cone of forces C_i that can be exerted on the edge E_i is defined by two vectors $\mathbf{n}_i + (\tan\alpha)\mathbf{t}_i$ and $\mathbf{n}_i - (\tan\alpha)\mathbf{t}_i$ where $\alpha \in [0, \pi/2)$ is the half-angle of the friction cone.

Since we will be dealing with two fingers that might not reside on the same polygon, we need two sets of entities for different polygons. Let all entities defined above correspond to the polygon P which is in contact with the first finger,

[1] $\mathbf{Z}_n$ is a group of non-negative integers less than n. Addition and subtraction are computed modulo n.

and let n', $\mathbf{v}'_i$, E'_i, $length'$, l'_i, L'_i, L', $\mathbf{t}'_i$, $\mathbf{n}'_i$ and C'_i be defined similarly for the polygon P' in contact with the second finger. If the two fingers are on the same polygon, then $n = n'$, $length = length'$, $L = L'$ and $X_i = X'_i$ where $X = \mathbf{v}, E, l, L, \mathbf{t}, \mathbf{n}$ or C.

The configuration space $\mathbf{C}$ of the two fingers is $[0, L) \times [0, L')$. Given a configuration $(u, u') \in \mathbf{C}$, we say that (u, u') composes a two-fingered grasp if and only if the two contact points $length^{-1}(u)$ and $(length')^{-1}(u')$ produce force closure. (Recall that $length$ is a function that maps a vertex into a number, so $length^{-1}$ gives a vertex.) The *graspable set* $G \subseteq \mathbf{C}$ is the set of all configurations that compose two-fingered grasps. *Graspable subsets* $G_{i,j}$ are graspable regions on edges E_i and E'_j defined by

$$G_{i,j} = G \cap ([L_i, L_{i+1}] \times [L'_j, L'_{j+1}]).$$

A. Computing $G_{i,j}$

Because each $G_{i,j}$ corresponds to configurations whose one finger is on E_i and the other is on E'_j, $G_{i,j}$ has been well-studied. According to Proposition 2, it has been shown in [13] that $G_{i,j}$ can be defined by eight linear inequalities in the parameters u and u'. However, there is an easier way to define $G_{i,j}$ as follows. Define the inverted force cone $-C'_j$ as $\{-\mathbf{x} \mid \mathbf{x} \in C'_j\}$. [3] showed that emptiness of $C_{i,j} = C_i \cap (-C'_j)$ implies emptiness of $G_{i,j}$. If $C_{i,j}$ is not empty, we claim that $G_{i,j}$ can be defined by no more than six points on the bounding rectangle.

Since a two-fingered grasp can be either *compressive*(squeezing grasp) or *expansive*(stretching grasp), we define for simplicity $DC_{i,j} = C_{i,j} \cup (-C_{i,j})$ as the double-sided cone of $C_{i,j}$ so that both the stretching and squeezing cases can be dealt with together. Now we prove the above claim by examining $DC_{i,j}$ centered on E_i. Let us first extend both sides of the edges E_i and E'_j to infinity, choose an arbitrary real number u, find $\mathbf{p}(u) = length^{-1}(u)$ on E_i, then let $DC_{i,j}(u)$ be the cone $DC_{i,j}$ centered at $\mathbf{p}(u)$. The intersection $I(u)$ of E'_j and the cone $DC_{i,j}(u)$ is a line segment on E'_j which represents the region that the second finger can be placed to achieve force closure with the first finger at $\mathbf{p}(u)$. This means for a given position of the first finger u, $length'(I(u))$ is the corresponding graspable interval in the second finger's configuration space (Fig. 1(a, b)).

It is easy to see that if u moves by Δu, $p(u)$ will move in the direction of $\mathbf{t}_i$ by the same distance Δu, the endpoints of $I(u)$ will move in the direction of $-\mathbf{t}'_j$ by the distance proportional to Δu, and the endpoints of $length'(I(u))$ move in the $-\Delta u$ direction by the distance proportional to Δu (Fig. 1(c, d)). These linear relationships imply that the graspable region is bounded by two straight lines. It is now obvious that cutting E_i and E'_j to their original lengths is equivalent to imposing four rectangular constraints $u \geq L_i$, $u \leq L_{i+1}$, $u' \geq L'_j$ and $u' \leq L'_{j+1}$ in the (u, u')-space (Fig. 1(e)). Therefore, $G_{i,j}$ can be defined with no more than six points on the bounding rectangle. In the real implementation, all defining points of $G_{i,j}$ can be found by computing endpoints

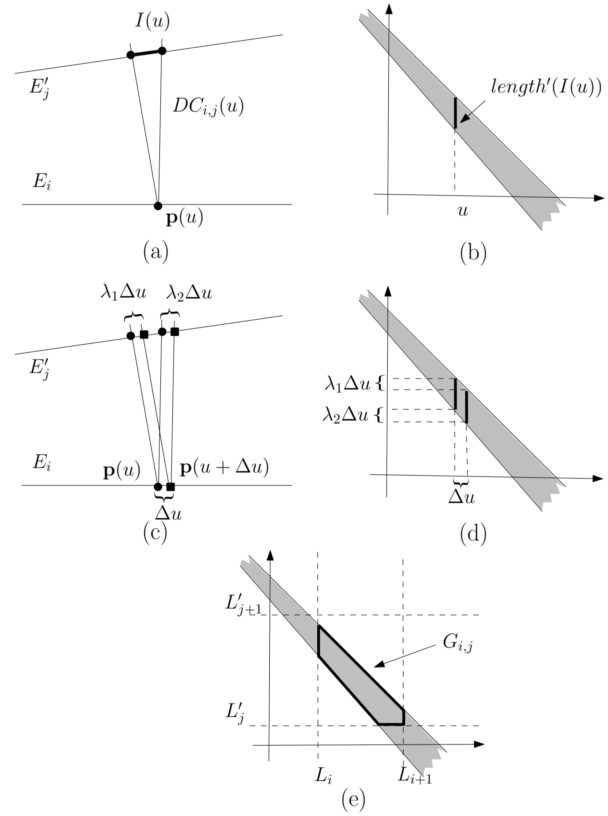

Fig. 1. (a) $I(u)$ is the graspable region of the second finger when the first finger is at $\mathbf{p}$. (b) $length'(I(u))$ is an interval in u' configuration space. (c) Endpoints of $I(u)$ move by the distances proportional to Δu. (d) Endpoints of $length'(I(u))$ move by the same distances as endpoints of $I(u)$, giving two straight lines bounding the graspable region. (e) $G_{i,j}$ is the result of cutting the infinite area by the rectangle.

of four intersections: $DC_{i,j}(\mathbf{v}_i) \cap E'_j$, $DC_{i,j}(\mathbf{v}_{i+1}) \cap E'_j$, $DC_{i,j}(\mathbf{v}'_j) \cap E_i$, and $DC_{i,j}(\mathbf{v}'_{j+1}) \cap E_i$ (Fig. 2).

B. Extending Configuration Space

An independent contact region when mapped to the configuration space becomes one, two, or four rectangles whose sides are parallel to u and u' axes. The region maps into one rectangle if it does not contain $\mathbf{v}_0$ or $\mathbf{v}'_0$, two rectangles if it contains $\mathbf{v}_0$ or $\mathbf{v}'_0$ but not both, or four rectangles if it contains both $\mathbf{v}_0$ and $\mathbf{v}'_0$ (Fig. 3). To eliminate the need to find independent contact regions with multiple rectangles, we extend the domain of $length^{-1}$ and $(length')^{-1}$ to the whole real line so they both become functions with periods L and L' respectively. ($length$ and $length'$ are no longer one-to-one.) The new G in the expanded configuration space $\Re^2$ can be defined from the old G with these periodic relations:

- $(u, u') \in G \Leftrightarrow (u + L, u') \in G$.
- $(u, u') \in G \Leftrightarrow (u, u' + L') \in G$.

We claim that despite infiniteness of G, every independent contact region has one corresponding rectangle within $G \cap [0, 2L] \times [0, 2L']$. This is easily proved by the following argument.

- Suppose a position of one finger is given, there must be some positions of the second finger that do not form force closure with the first finger.

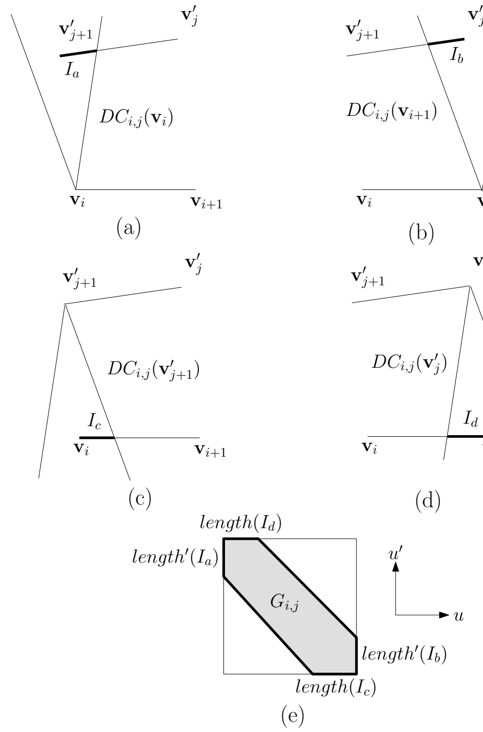

Fig. 2. Four intersections I_a, I_b, I_c and I_d are shown in (a), (b), (c) and (d). $G_{i,j}$ can be immediately defined by these intersections as shown in (e).

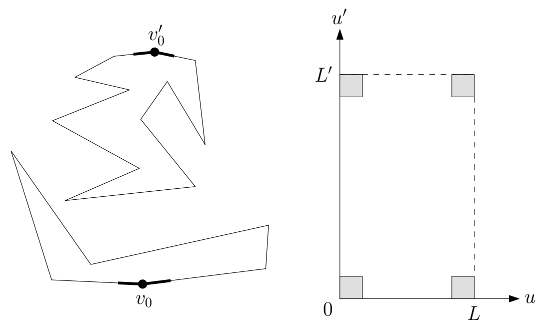

Fig. 3. The independent contact region shown in thick lines (left) maps to four rectangles in the configuration space (right).

- It follows that all u-constant line segments in G are shorter than L' and all u'-constant line segments in G are shorter than L.
- Since every independent contact region can be mapped into some rectangles in G whose sides are axis-parallel segments in G, one of these rectangles must lie inside $[0, 2L] \times [0, 2L']$.

The special case where the two fingers touch the same polygon can be handled with a smaller configuration space. G will be symmetric about the axis $u = u'$, which means we can cut out one half of G that lies above (or below) the line $u = u'$ (Fig. 4(a)). The remaining part of G above $u' > L$ (or to the right of $u > L$) can also be eliminated because to every rectangle crossing the line $u' = L$ (or the line $u = L$), there corresponds a rectangle in $[0, 2L] \times [0, L]$ (or $[0, L] \times [0, 2L]$) that represents the same configurations (Fig. 4(b)). Finally, the region to the right of the line $u = u' - L$ (or above the line $u = u' + L$) is redundant because no point on this line is in G (Fig. 4(c)). Therefore, the region of consideration is the shaded portion as shown in Fig. 4(d).

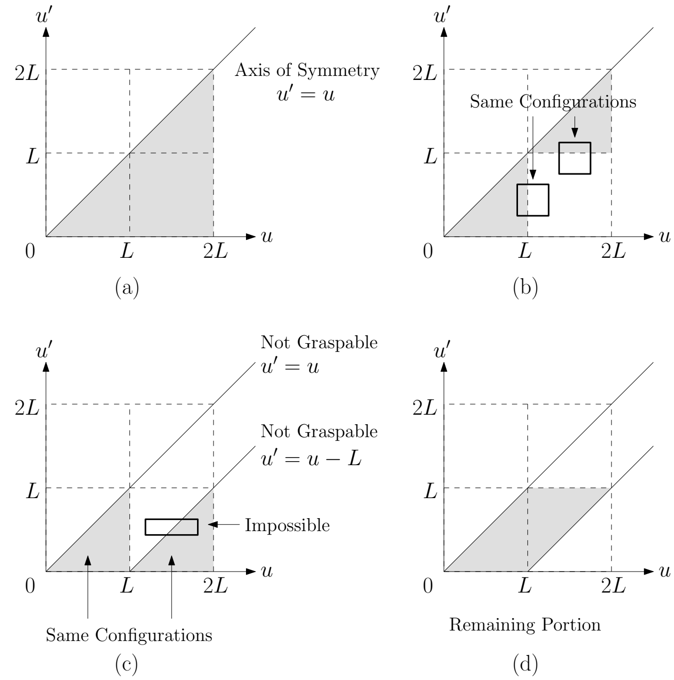

Fig. 4. (a) The axis of symmetry is $u' = u$. (b) For each rectangle that crosses the line $u' = L$, there corresponds another rectangle in $[0, 2L] \times [0, L]$ that crosses the line $u = L$. (c) The line $u' = u$ and $u' = u - L$ never intersect G, and the part of G to the right of $u' = u - L$ represents the same configurations as the remaining portion in $[0, L] \times [0, L]$. (d) The remaining portion to consider is shown in the shaded area.

C. Constructing G

Now we know that each $G_{i,j}$ contains at most six defining vertices, so all $G_{i,j}$ can be constructed within $O(nn')$ time. In the final algorithm, we will need the polygonal representation of G, so adjacent $G_{i,j}$ must be merged together into big pieces. We might need many simple polygons to define G because G does not have to be simple nor connected.

A vertex of some $G_{i,j}$ is a *defining vertex of* G, or *defines* G, if it is a vertex on a boundary of G. It can be observed that a vertex $\mathbf{v}$ of some $G_{i,j}$ defines G if and only if one of the following is true:

- $\mathbf{v}$ is not at a corner of the bounding rectangle.
- $\mathbf{v}$ is a corner of four bounding rectangles (one contains $G_{i,j}$ and the other three are adjacent), but is not contained in some $G_{k,l}$ bounded by these rectangles.

Note that if $G_{i,j}$ is neither empty nor full ("full" means $G_{i,j} = [L_i, L_{i+1}] \times [L'_j, L'_{j+1}]$), at least one vertex of $G_{i,j}$ must be a defining vertex of G.

The algorithm to find all simple polygons that define boundaries of G is described as follows. Let us first attach a state "used/unused" to all vertices of all $G_{i,j}$. All vertices are initialized as "unused". We scan through all values of i and j, and do the following:

- While $G_{i,j}$ has an "unused" vertex $\mathbf{v}$ that defines G,

- It is clear that $\mathbf{v}$ is on a boundary of G, so we can *trace* the boundary of G from $\mathbf{v}$ until we get back at $\mathbf{v}$.
- All vertices traced along the way defines a simple polygon which is a boundary of G. Mark these vertices as "used".

Note that *tracing* the boundary of G from $G_{i,j}$ may involve many $G_{k,l}$.

The tracing process can be simplified by first defining adjacencies of vertices that define G. Situations when two vertices $\mathbf{v}_1$ and $\mathbf{v}_2$ *that define* G are adjacent in G are listed below:

- If $\mathbf{v}_1$ and $\mathbf{v}_2$ are adjacent in the polygonal representation of $G_{i,j}$ and they lie on different sides of the bounding rectangle of $G_{i,j}$, they are adjacent in G.
- If $\mathbf{v}_1$ and $\mathbf{v}_2$ are adjacent in the polygonal representation of $G_{i,j}$ and they lie on the same side of the bounding rectangle of $G_{i,j}$, we assume without loss of generality that $\mathbf{v}_1, \mathbf{v}_2 \in \{L_i\} \times [L'_j, L'_{j+1}]$. $\mathbf{v}_1$ and $\mathbf{v}_2$ will be adjacent in G if $G_{i-1,j} \cap \overline{\mathbf{v}_1\mathbf{v}_2} = \varnothing$.
- If $\mathbf{v}_1$ and $\mathbf{v}_2$ belong to different pieces, i.e. $G_{i,j}$ and $G_{k,l}$, we assume without loss of generality that $\mathbf{v}_1 \in G_{i,j}, \mathbf{v}_2 \in G_{i-1,j}$. $\mathbf{v}_1$ and $\mathbf{v}_2$ can be adjacent in G if and only if they lie on the same segment $\{L_i\} \times [L'_j, L'_{j+1}]$ and
  - $\overline{\mathbf{v}_1\mathbf{v}_2} \subseteq G_{i-1,j}$ and $\overline{\mathbf{v}_1\mathbf{v}_2} \cap G_{i,j} = \{\mathbf{v}_1\}$, or
  - $\overline{\mathbf{v}_1\mathbf{v}_2} \subseteq G_{i,j}$ and $\overline{\mathbf{v}_1\mathbf{v}_2} \cap G_{i-1,j} = \{\mathbf{v}_2\}$.

IV. Optimal Independent Contact Regions

With the polygon G in hand, we can find the *best* independent contact region. In this paper, we study two meanings of *best* defined by two objective functions:

1) ab
2) $\min\{a, b\}$

where a and b are side lengths of an independent contact region. (Recall that independent contact regions are rectangles that lie in G.)

The first criterion, introduced in [13], defines the best independent contact region as the rectangle with greatest area, while the other criterion, introduced in [4], prefers the rectangle with greatest inscribed square. Optimizing the second objective function is equivalent to finding the largest axis-parallel square inscribed in G, so we name this problem *The Largest Square Independent Contact Region Problem.* The problem of optimizing the first criterion is accordingly named *The Largest Rectangular Independent Contact Region Problem.*

A. *The Largest* RECTANGULAR *Independent Contact Region Problem*

The problem of finding the largest rectangle inscribed in a polygon (with or without holes) has been thoroughly studied by Karen Daniels *et al.* in [15]. If that polygon has m vertices, the algorithm can find the largest rectangle within $O(m \log^2 m)$ time bound.

We can supply vertices of G directly to the algorithm and obtain the solution within $O(m \log^2 m) = O(nn' \log^2(nn'))$. We have experimented some examples shown in Fig. 5. The method of Karen Daniels *et a.* is very complex to implement. Therefore, dynamic programming technique is applied instead. The dynamic program memorizes the bounds of a rectangle which is possibly optimum. The bounds are adjusted while G is traversed.

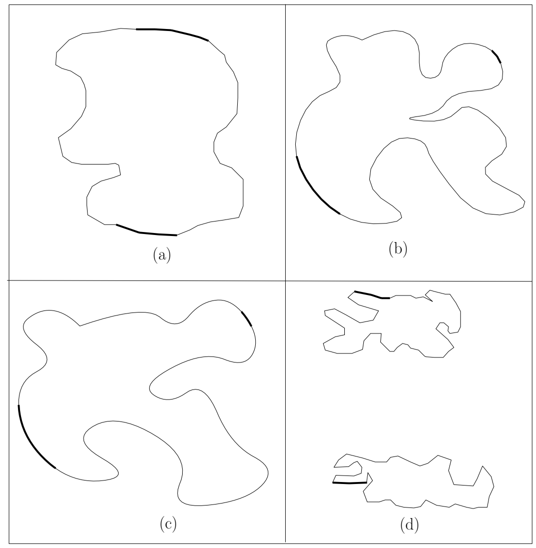

Fig. 5. Different test objects supplied to the algorithm that maximizes the first criterion. Thick lines represent the optimal independent contact region when the half-angle of the friction cone is 20°. (a) The input object has 62 vertices. (b) The input object has 128 vertices. (c) The input object has 256 vertices. (d) The input object has 112 vertices.

B. *The Largest* SQUARE *Independent Contact Region Problem*

To solve this problem, we start by computing the L_∞ Voronoi diagram [16] inside G. We claim that the largest square in G must be centered at a Voronoi vertex. This is justified by the following argument: Let $\mathbf{v}$ be a point in G and let $square(\mathbf{v})$ denote the largest square in G centered at $\mathbf{v}$.

- If $\mathbf{v}$ is inside a Voronoi region (or site), $square(\mathbf{v})$ must have one corner on an edge of G. Moving $\mathbf{v}$ *away* from that edge will increase the size of $square(\mathbf{v})$. We can move $\mathbf{v}$ in such direction until it reaches a Voronoi edge or a Voronoi vertex while $square(\mathbf{v})$ is growing.
- If $\mathbf{v}$ is on a Voronoi edge, $square(\mathbf{v})$ has two corners on two edges of G. If these edges are parallel, the Voronoi edge must also be parallel to them, and we can move $\mathbf{v}$ in either direction along the Voronoi edge until it hits a Voronoi vertex without changing the size of $square(\mathbf{v})$. But if the two edges are not parallel, it is always possible to define the direction along the Voronoi edge that brings $\mathbf{v}$ *away* from the two edges. Moving $\mathbf{v}$ in this *away* direction, the size of $square(\mathbf{v})$ is increasing along the way, and $\mathbf{v}$ will eventually coincide with a Voronoi vertex.

Therefore, we only need to check squares centered at Voronoi vertices to find the largest square inscribed in G.

E. Papadopoulou and D. T. Lee presented in [16] an algorithm to construct the L_∞ Voronoi diagram of a polygon (with or without holes) with m vertices within $O(m \log m)$ time bound and showed that the number of Voronoi vertices is $O(m)$. When a Voronoi vertex is generated by the algorithm, the size of the square centered at that vertex is known. As a result, the largest square inscribed in the polygon can be found in $O(m)$ time after all Voronoi vertices have been found.

By supplying G to the algorithm, we need $O(m \log m) = O(nn' \log(nn'))$ time to construct the L_∞ Voronoi diagram and $O(m) = O(nn')$ to search for the largest square. The overall running time is therefore $O(nn' \log(nn'))$. However, the algorithm of constructing L_∞ Voronoi diagram have not been implemented. Some preliminary results shown in Fig. 6 are obtained by computing dynamic program in the same fashion as maximizing the largest rectangular independent contact regions.

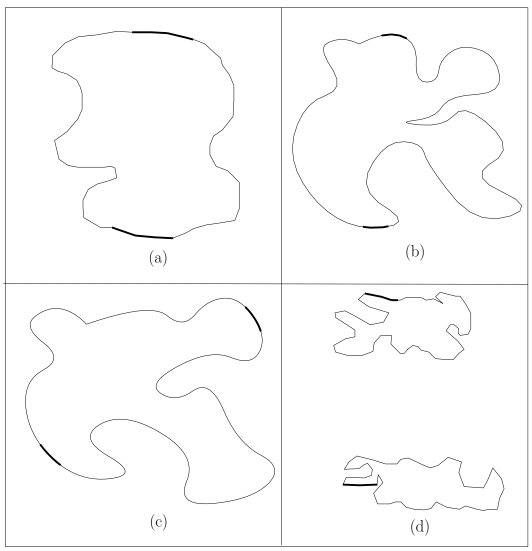

Fig. 6. Different test objects supplied to the algorithm that maximizes the second criterion. Thick lines represent the optimal independent contact region when the half-angle of the friction cone is 20°. (a) The input object has 62 vertices. (b) The input object has 128 vertices. (c) The input object has 256 vertices. (d) The input object has 112 vertices.

The Voronoi diagram can also be used to measure the quality of a given grasp. If we are given $(u, u') \in G$, $square(u, u')$ can be computed, and its size can be used as an indicator of the ability to preserve force closure under perturbation.

V. Conclusion and Future Work

We are concerned with the problem of determining the optimal independent contact region for two-fingered grasps of a 2D object with Coulomb friction. The configuration space of the two fingers is $\Re^2$ and independent contact regions are rectangles whose sides are parallel to the two axes. Two reasonable definitions of the *best* independent contact region are given by Ponce and Faverjon in [13] and [4]. We presented two methods to find the optimal independent contact region for two optimization criteria. The first criterion defines the *largest rectangular independent contact region problem* (discussed in section IV-A) that can be solved in $O(n^2 \log^2 n)$ time complexity, while the second criterion defines the *largest square independent contact region problem* (discussed in section IV-B) that can be solved in $O(n^2 \log n)$ provided that the input object is defined by simple polygons with n total vertices. Both methods can yield independent contact regions across edges.

Extension to three-fingered grasping problems will be simple if there exists a representation of graspable volume which is simple enough. Once we have all $G_{i,j,k}$, G can be constructed, and we can work on it. Though an analytical algorithm for optimizing both criteria is not known yet, the idea to construct a discretized version of 3D generalized Voronoi diagram using graphics hardware by Hoff *et al.* presented in [17] can be modified to produce discretized L_∞ Voronoi diagrams of three dimensional shapes.

References

[1] J. Salisbury, "Kinematic and force analysis of articulated hands," Ph.D. dissertation, Stanford University, Stanford, CA, 1982.

[2] B. Mishra, J. Schwartz, and M. Sharir, "On the existence and synthesis of multifinger positive grips," *Algorithmica, Special Issue: Robotics*, vol. 2, no. 4, pp. 541–558, November 1987.

[3] V.-D. Nguyen, "Constructing force-closure grasps," *International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, June 1988.

[4] J. Ponce and B. Faverjon, "On computing three-finger force-closure grasps of polygonal objects," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 868–881, December 1995.

[5] J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet, "On computing four-finger equilibrium and force-closure grasps of polyhedral objects," *International Journal of Robotics Research*, vol. 16, no. 1, pp. 11–35, February 1997.

[6] A. Blake, "A symmetry theory of planar grasp," *International Journal of Robotics Research*, vol. 14, no. 5, pp. 425–444, 1995.

[7] C. Ferrari and J. Canny, "Planning optimal grasps," in *IEEE Int. Conf. on Robotics and Automation*, Nice, France, June 1992, pp. 2290–2295.

[8] X. Zhu and J. Wang, "Synthesis of force-closure grasps on 3-d objects based on the q distance," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 669–679, 2003.

[9] H. D. X. Zhu and S. K. Tso, "A pseudodistance function and its applications," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, p. 344352, 2004.

[10] C. P. Tung and A. C. Kak, "Fast construction of force-closure grasps," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, p. 615626, 1996.

[11] J. Cornella and R. Suarez, "Determining independent grasp regions on 2d discrete objects," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005.

[12] N. S. Pollard, "Closure and quality equivalence for efficient synthesis of grasps from examples," *International Journal of Robotics Research*, vol. 23, no. 6, pp. 595–614, June 2004.

[13] B. Faverjon and J. Ponce, "On computing two-finger force-closure grasps of curved 2D objects," in *IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, April 1991, pp. 424–429.

[14] J. Cornella and R. Suarez, "Fast and flexible determination of force-closure independent regions to grasp polygonal objects," in *IEEE Int. Conf. on Robotics and Automation*, 2005.

[15] V. M. K. Daniels and D. Roth, "Finding the largest area axis-parallel rectangle in a polygon," *CGTA: Computational Geometry: Theory and Applications*, vol. 7, 1997.

[16] E. Papadopoulou and D. T. Lee, "The l_∞-voronoi diagram of segments and vlsi applications." *Int. J. Comput. Geometry Appl.*, vol. 11, no. 5, pp. 503–528, 2001.

[17] K. E. Hoff III, J. Keyser, M. Lin, D. Manocha, and T. Culver, "Fast computation of generalized Voronoi diagrams using graphics hardware," *Computer Graphics*, vol. 33, no. Annual Conference Series, pp. 277–286, 1999.