UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Grasp synthesis for 3D objects

Máximo Alejandro Roa Garzón, Raúl Suárez Feijoo

IOC: Divisió de Robòtica

IOC-DT-P-2006-23
Juliol 2006

Institut d'Organització i Control
de Sistemes Industrials

GRASP SYNTHESIS FOR 3D OBJECTS[1]

Máximo Alejandro Roa Garzón[2]
Raúl Suárez Feijóo

Institute of Industrial and Control Engineering (IOC)
Technical University of Catalonia (UPC)
Barcelona, Spain

July 2006

[1]This work was partially supported by the CICYT projects DPI2004-03104 and DPI2005-00112

[2]Department of Mechanical and Mechatronic Engineering - National University of Colombia; at present with the IOC-UPC, supported by the Programme Alβan (European Union Programme of High Level Scholarships for Latin America), Scholarship No. E04D039103CO.

Contents

Chapter 1

Basic background

Robotic hands are flexible and powerful tools that can provide industrial and service robots with a high ability to manipulate objects; however, an appropriate grasp planning is required in order to grasp and manipulate objects in the real world. A grasp planner calculates the position of the fingers on the object surface, while fulfilling a basic grasp property. In general, grasps are computed to ensure the immobility of the object in front of external disturbances, satisfying one of the following properties: form-closure (or complete kinematical restriction), when the position of the fingers ensure the object immobility, or force-closure (hereafter FC), when the forces applied by the fingers ensure the object immobility [1]. The property to be used largely depends on the field of application: form-closure is used when the task requires a robust grasp not relying on friction, e.g. the fixture of objects to be manufactured or inspected, while force-closure is specially used in grasping and manipulation of objects with a low number of frictional contacts, using for instance mechanical grippers or hands.

1.1 Generalized forces

Consider a coordinate system located at the object center of mass (CM) to describe the positions $\boldsymbol{p}$ of the contact points and the forces applied on the grasped object. A force $\boldsymbol{f}_i$ applied on the object at the point $\boldsymbol{p}_i$ generates a torque $\boldsymbol{\tau}_i = \boldsymbol{p}_i \times \boldsymbol{f}_i$ with respect to CM. The force and the torque are grouped together in a wrench (also known as generalized force vector) given by

$$\tilde{\boldsymbol{\omega}}_i = \begin{pmatrix} \boldsymbol{f}_i \\ \boldsymbol{\tau}_i \end{pmatrix} = \alpha_i \begin{pmatrix} \hat{\boldsymbol{n}}_i \\ \boldsymbol{p}_i \times \hat{\boldsymbol{n}}_i \end{pmatrix} \tag{1.1}$$

$\boldsymbol{\omega} \in \Re^d$, where d is the dimension of the wrench space; $d = 3$ for the two-dimensional physical space (planar problem with 2D objects) and $d = 6$ for 3D objects in the three-dimensional physical space.

1.2 Number of fingers required to grasp objects

The number of fingers required to get a FC grasp depends on the type of contact between the fingertip and the object. There are three basic contact types: frictionless point contact (FPC), point contact with friction (PCWF, also known as hard finger contact) and soft finger contact (SC). Frictionless grasps are also known as positive grips; upper and lower bounds in the number of fingers required for a FC grasp in two and three-dimensional objects are computed in [2] using two classic theorems in convex analysis: Caratheodory and Steinitz theorems. An algorithm to find at least one frictionless FC grasp for polyhedral objects is presented in [2]; the algorithm is linear in time according to the number of faces of the polyhedron. An object with rotational symmetry (exceptional object) does not have a FC grasp with frictionless contacts.

Table 1.1: Bounds in the number of fingers required to find a FC grasp.

Object	Object type	FPC	PCWF	SC
Two-dimensional	Exceptional	np	3	3
(d=3)	Non exceptional	4	3	3
Three-dimensional	Exceptional	np	4	4
(d=6)	Non exceptional	7	4	4

np: not possible

Frictional and frictionless contacts can also be studied from a geometrical point of view [3]. For frictional contacts (PCWF and SC), 3 and 4 fingers are required to assure a FC grasp in *any* 2D or 3D object, respectively, as shown in Table 1.1 (although for some objects, FC grasps with a smaller number of fingers can be achieved).

1.3 Grasp planning for three-dimensional objects

Grasp planning, also known as grasp synthesis, has been tackled with two different approaches: empirical or analytical. The empirical (knowledge-based) approach imitates human grasp using heuristics (constructive algorithms) to choose a grasp shape from a set of basic hand postures that depend on the task and on the object's geometry. On the other hand, analytic approaches choose the finger positions and the hand configuration with kinematical and dynamical formulations, in general optimizing an objective function such as the grasp stability or the resistance to external perturbations.

Different algorithms have been developed in grasp planning for two-dimensional objects (either polygonal [4, 5], non-polygonal [6] or discrete objects [7]); however, grasp synthesis in three-dimensional objects is still an active research area, mainly due to the complex geometry and high dimensionality of the grasp space. For instance, two coordinates are required to represent a grasp point on the surface of a 3D object; therefore, the computation of a FC grasp implies searching for a solution in a space of dimension $2n$ (at least), with n being the number of contacts.

This technical report reviews the literature for analytical grasp synthesis in polyhedral (Chapter 2) and general 3D objects (Chapter 3); Chapter 2 complements a previous IOC technical report [8] on the same subject with new references. Chapter 4 reviews some works related to three-dimensional grasp synthesis; Section 4.1 presents the main empirical approaches for grasp synthesis, and Section 4.2 shows the relation between grasp synthesis and fixture design for manufacturing tasks.

Chapter 2

Grasp synthesis for polyhedral objects

Most of the works in grasp synthesis for 3D objects generate grasps for polyhedral objects, composed of a finite number of flat faces. The normal direction to the surface of the object is the same for all of the points in the same face of the polyhedron, thus facilitating the development of analytical algorithms to generate a grasp for the object.

2.1 Two-fingers grasps

Initial works in grasp synthesis for polyhedral objects considered two-fingered grasps with soft contacts [9, 10], searching for independent contact regions. If each finger is in its independent contact region, a FC grasp is always obtained; thus, the use of independent contact regions makes the grasp more robust in front of possible changes in contact placement.

As polyhedra have faces with constant normal directions, the FC problem can be split into two subproblems:

- Force-direction closure: verifies that the friction cones from the finger contacts span all the directions in R^3. Figure 2.1 shows two soft finger contacts on two faces of a polyhedron, with ψ being the angle between the two planes. Positive combination of the two friction cones spans any direction in R^3 iff $\psi < 2\phi$, with $\mu = \tan\phi$ being the friction coefficient.

- Torque closure: tests that the positive combinations of the contact forces generates all the possible torques. This is achieved if the segment P_1P_2 (or P_2P_1) joining the two contact points P_1 and P_2 is completely included inside the friction cones generated in P_1 and P_2.

The torque-closure condition is a necessary and sufficient condition for the existence of a FC grasp with two soft fingers. Note that a FC grasp with two point contacts with friction (PCWF) can not exist, since it could not resist torques around the line joining the two contact points.

The force-direction closure condition is verified with a simple test on the angle between the normal directions to the grasped faces. After a pair of faces satisfying this condition is found, the two friction cones overlap in a range given by $C_{P_2} \cap -C_{P_1}$; the overlapping sector is approximated with the maximum cone that lies inside the sector (in the original algorithm, the polyhedron faces are approximated by circumscribed circular discs to simplify this step). The vertex I of the double cone with angle $\pm(C_{P_2} \cap -C_{P_1})$ is placed in the bisector plane of the two faces; the intersection between the two faces and the double cone produces the independent contact regions, a_1 and a_2. The vertex I is placed in such a way that maximizes the region´s area (Figure 2.1).

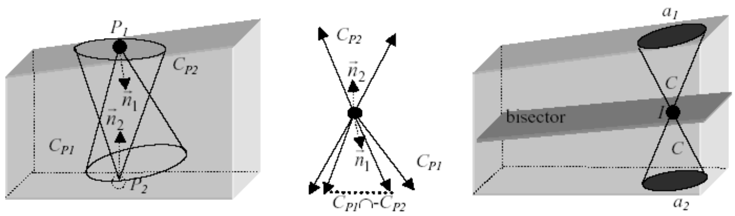

Figure 2.1: Grasp with two soft contact fingers (Source: [8]).

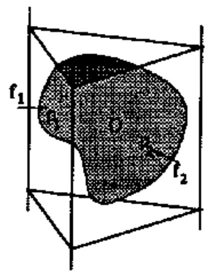

Figure 2.2: Optimum three-finger grasp given by the maximum circumscribed prism in the object (Source: [11]).

A classification of three-finger grasps in four basic groups, depending on the number of overlapped friction cones, is also presented in [10], as well as the conditions that the wrenches must fulfill in order to get form-closure grasps with 7 frictionless fingers.

2.2 Three-fingers grasps

The computation of three-finger optimum FC grasps is tackled in [11], using an optimality criterion based on decoupled wrenches. First of all, grasps that best resist external forces in the grip plane (the plane determined by the three grasp forces, that must be concurrent) are chosen, and among these grasps those that best resist the applied torques perpendicular to the grip plane are considered to be optimum. In this way, the optimum grasp for a smooth object is the grasp determined by the maximum circumscribing prism, as shown in Figure 2.2. In the particular case of polyhedra, the grasp points should be placed on the polyhedron vertices under this criterion; thus, grasp synthesis is reduced to test all the possible vertices combinations, so the algorithm complexity is $O(n^3)$ with n the number of faces of the polyhedral object. As a particular case, if the resistance to moments in the grip plane should be optimized, then the optimum solution is given by the maximum equilateral triangle inscribed in the object.

Several necessary and sufficient conditions for three-finger FC grasps are presented in [12] (Figure 2.3):

- Necessary condition: there exists a point in the intersection of the plane formed by the three contact points with the double-sided friction cones at these points.
- Sufficient condition: there exists a point in the intersection of the three open internal friction cones with the triangle formed by these contact points.

The internal (or negative) friction cones are those pointing inside the object; the external (or positive) friction cones are those pointing outside the object. The previous sufficient

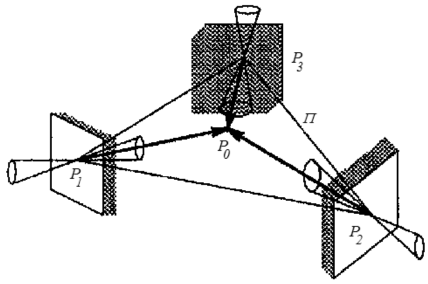

Figure 2.3: Grasping a polyhedron with three frictional fingers (Source: [12]).

condition is very restrictive, as there may be several grasps whose intersecting point does not lie in the internal friction cones. Therefore, the following condition is proposed

- Sufficient condition: three points P_1, P_2 and P_3 form a FC grasp if the pairwise angles between the internal normals at these points are less than $\pi - 2\theta$, there exist three vectors making angles less than $\theta/3$ with these normals and positively spanning some fixed vector plane Π, and there exists a point P_0 such that for $i = 1, 2, 3$ the vector $\boldsymbol{P_0} - \boldsymbol{P_i}$ lies in the intersection of the plane Π with the double-sided friction cones of half-angle $\theta/3$ in P_i.

For a fixed vectorial plane Π, the condition can be decomposed into eight linear sufficient conditions depending on whether P_0 lies in the internal or external friction cone at each contact point P_i. Moreover, the plane containing the three contact points is not fixed, but it is parallel to Π. Each contact point may be parametrized with two variables (u_i, v_i) defining its position in the plane of the corresponding face; besides, a set of linear restrictions in (u_i, v_i) can be used to specify that P_i lies in a convex face. The problem solution must find 9 variables: 6 defining the position of the contact points P_i and 3 defining the position of the intersection point P_0. To simplify the problem, P_0 may be eliminated from the formulation, and the 6 remaining variables placing the contact points on the object can be found with linear programming.

2.3 Four-fingers grasps

2.3.1 Concurrent grasps

An algorithm to find four-finger grasps is proposed in [13], complementing the previous work in [12] by the same authors. The necessary and sufficient condition for four non coplanar points to form an equilibrium grasp with four nonzero contact forces is:

P1: there exist four lines in the corresponding double-sided friction cones that intersect in a single point, form two flat pencils having a line in common but lying in different planes, or form a regulus (Figure 2.4), and

P2: the vectors parallel to these lines and lying in the internal friction cones at the contact points positively span R^3.

The previous condition allows the computation of equilibrium grasps; in the presence of friction, a sufficient condition to get a 3D FC grasp is non-marginal equilibrium, i.e. the applied forces must be inside the open friction cones at the contact points. For convenience, the equilibrium conditions should be linear in the unknown grasp parameters (finger positions and contact forces), so the problem can be solved with linear programming; however, P2 is a nonlinear condition. To obtain linear conditions, a new sufficient condition for four non coplanar points to form an equilibrium grasp with four nonzero contact forces is proposed (Figure 2.5):

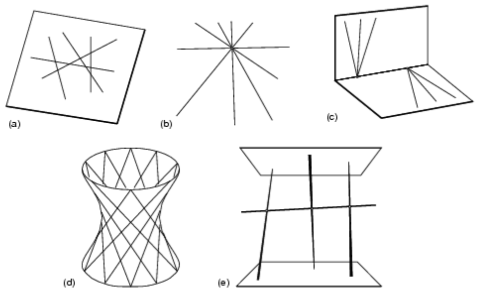

Figure 2.4: Configurations of positively dependent lines: a) Coplanar lines, b) Concurrent lines, c) Flat pencils, d) Regulus, e) A regulus is the geometric place of the lines that simultaneously intersect three skew lines (they do not cross or intersect) (Source: [13]).

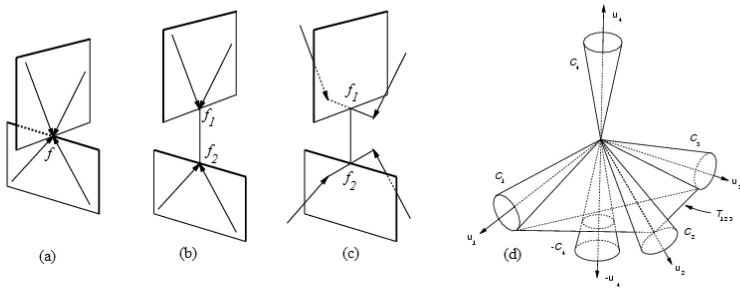

Figure 2.5: Sufficient condition for a 4-finger equilibrium grasp; proposition P1 demands four lines that a) be concurrent, b) form two noncoplanar flat pencils or c) form a regulus; d) illustrates proposition P3 (Source: [13]).

P1: there exist four lines in the corresponding double-sided friction cones that intersect in a single point, form two flat pencils having a line in common but lying in different planes, or form a regulus, and

P3: the surface normals at the four contact points θ-positively span R^3. Four vectors θ-positively span R^3 when, for any triplet $\boldsymbol{u_1}, \boldsymbol{u_2}, \boldsymbol{u_3}$ of these vectors, the cones C_1, C_2, C_3 of half-angle θ centered in $\boldsymbol{u_1}$, $\boldsymbol{u_2}$ and $\boldsymbol{u_3}$ lie in the interior of the same half-space, and the cone C_4 of half-angle θ centered on the opposite direction to the fourth vector $\boldsymbol{u_4}$ lies in the interior of the intersection of the trihedra formed by all the triplets of vectors belonging to C_1, C_2 y C_3.

The advantage of proposition P3 is that it depends only on the normal direction to the grasped faces, while P2 depends on the real directions of the applied forces. This allows the algorithm implementation in two steps: first, faces satisfying P3 are selected, and then the configurations satisfying P1 are computed. Condition P1 is decomposed in 16 elementary conditions in the case of concurrent grasps (the intersection point may lie in the internal or external friction cone at each contact point), so linear programming is used to search the FC grasps. The algorithm implemented in [13] is restricted to the synthesis of four finger concurrent grasps, and has the following steps:

1. Search of faces satisfying condition P3

2. Linear restrictions to define stable grasp regions
   Each face F_i is defined parametrically by $\boldsymbol{x_i} = \boldsymbol{x_{0i}} + a_i \boldsymbol{u_i} + b_i \boldsymbol{v_i}$, with $(\boldsymbol{u_i}, \boldsymbol{v_i})$ a vector basis for the plane of F_i. If F_i is bounded by n_i edges, the parameters a_i, b_i must also satisfy n_i linear constraints with general form

$$f_{ij}(a_i, b_i) \leq 0, \ j = 1, ..., n_i \tag{2.1}$$

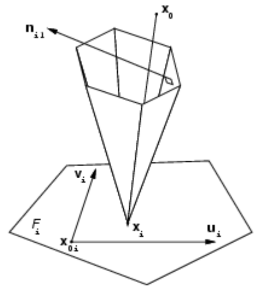

Figure 2.6: Location of the contact point and approximation of the friction cone (Source: [13]).

meaning that $\boldsymbol{x_i}$ must lie within the face F_i.
The internal friction cone C_i associated with the face F_i is approximated by an m-sided pyramid, whose faces have internal normals $\boldsymbol{n_{ij}}$, with $j = 1, ..., m$ (Figure 2.6). A point $\boldsymbol{x_0} = (x_0, y_0, z_0)$ belongs to C_i if the following restrictions are satisfied

$$(\boldsymbol{x_0} - \boldsymbol{x_{0i}} - a_i\boldsymbol{u_i} - b_i\boldsymbol{v_i}) \cdot \boldsymbol{n_{ij}} \geq 0,\ j = 1, ..., m \tag{2.2}$$

There are $4m + \sum_{i=1}^{n} n_i$ constraints associated with the four grasping points, defining a polytope in R^{11} (there are 2 variables to locate each contact point and 3 variables to locate the concurrence point). To consider all the possible combinations of internal and external friction cones providing a solution to P1, a series of equations such as (2.2) are used, except that the inequality sign is reversed for the external cones. The total set of solutions is the union of the polytopes corresponding to the different combinations.

3. Elimination of the concurrence point
The optimization of an objective function, subject to constraints (2.1) and (2.2), is achieved with linear programming. However, the main variables of the problem are a_i, b_i; the concurrence point (x_0, y_0, z_0) has a minor importance, i.e. it is more important to characterize the equilibrium regions in the 8-dimensional configuration space of the parameters a_i, b_i, defining the grasp configuration. This implies that the variables (x_0, y_0, z_0) must be eliminated from the restrictions, or, equivalently, the polytope defined by (2.1) and (2.2) in an 11-dimensional space must be projected in an 8-dimensional space. This projection defines a new set of constraints (2.1) and (2.2'), with (2.2') representing the constraints (2.2) modified to eliminate the concurrence point. To achieve this purpose, two algorithms are proposed: one based on the gaussian elimination of the undesired variables, and another one based on the contour tracking of the polytope in the projected space.

4. Computation of the maximal independent contact regions
Independent contact regions are computed to minimize the grasp sensitivity to errors in finger´s positions. In the grasp configuration space, these regions are parallelepipeds with sides aligned with the coordinate axes, and contained in the polytope of equilibrium grasps. The algorithm searches for maximal contact regions, i.e. regions maximizing the minimum of the lengths of the parallelepiped edges. A parallelepiped in the configuration space is defined by $R_i = \left[a_i^-, a_i^+\right] \times \left[b_i^-, b_i^+\right]$, $i = 1, ..., 4$. Let u be the minimum of the lengths of the corresponding intervals (i.e. the variable to be maximized); to find the maximal contact regions, the following constraints must be added to the initial ones (2.2'):

$$\begin{cases} u \geq 0 \\ u \leq a_i^+ - a_i^-,\ i = 1, ..., 4 \\ u \leq b_i^+ - b_i^-,\ i = 1, ..., 4 \end{cases} \tag{2.3}$$

expressing that u is positive and smaller than the length of each interval. In general, there will be multiple optimal solutions, so a secondary criterion is required to find a unique solution: the algorithm tries to center the center of mass of the object in the tetrahedron formed by the contact points. To implement this criterion, a new variable v must be introduced, measuring the distance L_∞ between the center of mass of the object $\boldsymbol{g_p} = (x_p, y_p, z_p)$ and the center $\boldsymbol{g_c} = (x_c, y_c, z_c)$ of the contacts corresponding to the centers of the rectangles R_i. Thus, $v = \max(|x_c - x_p|, |y_c - y_p|, |z_c - z_p|)$, yielding to the following restrictions:

$$\begin{cases} v \geq x_p - x_c \\ v \geq x_c - x_p \\ v \geq y_p - y_c \\ v \geq y_c - y_p \\ v \geq z_p - z_c \\ v \geq z_c - z_p \end{cases} \tag{2.4}$$

In summary, the search of the maximal independent contact regions is solved with linear programming: the constraints are (2.1), (2.2') (written once for each one of the 256 vertices of the parallelepiped), (2.3) and (2.4), and the objective function to be maximized is a weighted combination $w_u u - w_v v$, with the weights w defined by the user. There are 18 variables: $a_i^-, a_i^+, b_i^-, b_i^+$ $(i = 1, ..., 4)$ with the auxiliary variables u and v. For each quadruple of faces, the representative grasp is selected as the center of the maximal independent contact region.

2.3.2 Flat pencil and regulus grasps

The previous work, [13], is focused on four finger concurrent grasps. However, flat pencil and regulus grasps may be useful in particular cases; for instance, in the manipulation of an elongated object for which concurrent grasps do not exist, two cooperating robots with simple (two-finger) grippers may be used, employing a flat pencil grasp. The sufficient condition previously presented for four noncoplanar points to form an equilibrium grasp (given by propositions P1 and P3) was originally exposed in [14] to search for independent contact regions with linear programming (using the simplex method) for concurrent and flat-pencil grasps, and to obtain at least one regulus grasp. The algorithms to search for these grasps are based on geometrical considerations, and they assume that four faces whose normals θ-positively span R^3 have been previously found; the success of the previous method depends on the initial selection of the grasp focuses, denoted as f_i in Figure 2.5.

Another geometrical approach to compute these kind of grasps is proposed in [15]. Initially, a set of possibly good faces is selected based on purely geometrical considerations, and then the grasping points on these faces are selected.

2.4 Grasps with any number n of fingers

Few works have tackled the grasp synthesis for any number of fingers. Some of these works deal with the problem of grasp synthesis for n fingers given the position for k of the fingers; this kind of problem arises frequently in fixture design (Section 4.2).

2.4.1 Given the location for $n-1$ fingers

The FC grasp synthesis problem for n fingers when $n-1$ fingers have fixed positions (and the grasp with the $n-1$ fingers is not FC) is discussed in [16]. The friction cone is approximated with an m-sided polyhedral cone (or convex pyramid) (Figure 2.7); the grasping force at the contact point is given by

$$\boldsymbol{f_i} = \sum_{j=1}^{m} u_{ij} \boldsymbol{s_{ij}}, \ u_{ij} \geq 0 \tag{2.5}$$

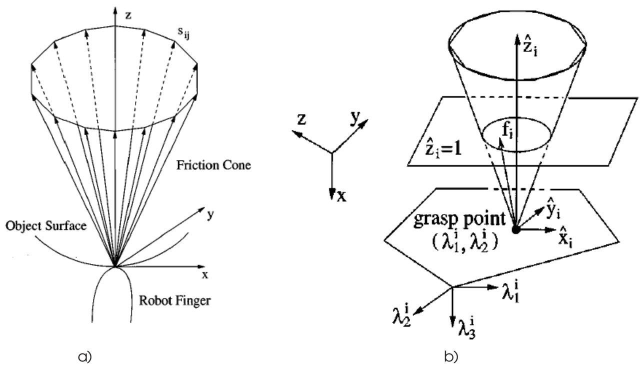

Figure 2.7: Grasp synthesis with n fingers: a) Approximation to the friction cone, b) Parametrization of an object face (Source: [16]).

with $\boldsymbol{s_{ij}}$ representing the vector of the j-th edge in the polyhedral cone; the vectors are normalized in a way such that $\boldsymbol{s_{ij}} \cdot \boldsymbol{n_i} = 1,\ i = 1, \ldots, n,\ j = 1, \ldots, m$. $\boldsymbol{n_i}$ is the unitary vector normal to the object surface at the point $\boldsymbol{p_i}$, and $\boldsymbol{p_i}$ is the position vector for the grasping point i with respect to a coordinate system located in the center of mass of the object. Therefore, $\sum_{j=1}^{m} u_{ij} = \boldsymbol{f_i} \cdot \boldsymbol{n_i}$, i.e. $\sum_{j=1}^{m} u_{ij}$ is the amplitude of the normal component for the contact force $\boldsymbol{f_i}$. The wrench produced by the force $\boldsymbol{f_i}$ is

$$\boldsymbol{\varpi_i} = \begin{pmatrix} \boldsymbol{f_i} \\ \boldsymbol{p_i} \times \boldsymbol{f_i} \end{pmatrix} = \sum_{j=1}^{m} u_{ij} \boldsymbol{g_{ij}},\ \boldsymbol{g_{ij}} = \begin{pmatrix} \boldsymbol{s_{ij}} \\ \boldsymbol{p_i} \times \boldsymbol{s_{ij}} \end{pmatrix} \tag{2.6}$$

The vectors $\boldsymbol{g_{ij}}$ are denoted as the primitive contact wrenches of the i-th finger.

The total wrench applied on the object by the fingers is

$$\boldsymbol{\omega} = \sum_{i=1}^{n} \boldsymbol{\varpi_i} = \sum_{i=1}^{n} \sum_{j=1}^{m} u_{ij} \boldsymbol{g_{ij}} = \boldsymbol{G} u \tag{2.7}$$

with

$$\boldsymbol{G} = (\boldsymbol{g_{11}}, \boldsymbol{g_{12}}, \ldots, \boldsymbol{g_{1m}}, \ldots, \boldsymbol{g_{n1}}, \boldsymbol{g_{n2}}, \ldots, \boldsymbol{g_{nm}}) \tag{2.8}$$

$$u = (u_{11}, u_{12}, \ldots, u_{1m}, \ldots, u_{n1}, u_{n2}, \ldots, u_{nm})^T \tag{2.9}$$

$\boldsymbol{G} \in \Re^{6 \times nm}$; $\boldsymbol{G}$ is called the wrench matrix, and its column vectors are the primitive contact wrenches. For convenience, $\boldsymbol{g_i}$ is used instead of $\boldsymbol{g_{ij}}$ to denote the i-th column vector of the wrench matrix $\boldsymbol{G}$, and u_i represents the i-th component of the vector u. The number of primitive contact wrenches is $N = nm$.

A n-fingers grasp is FC if for any external wrench $\boldsymbol{\omega}$ applied on the object, it is always possible to find an u with all $u_i > 0$ such that $\boldsymbol{G}u = -\boldsymbol{\omega}$. This condition is equivalent to that the convex hull $CH(\boldsymbol{G})$ of the primitive contact wrenches $\boldsymbol{\omega_i}$ contains the neighborhood of the origin in the wrench space $\Re^6$.

To represent the contact point of the finger i on the polyhedron face, a local coordinate system $\left\{\lambda_1^i, \lambda_2^i, \lambda_3^i\right\}$ is attached to the face; the origin of the system is located at one vertex and the axis λ_3^i is parallel to the normal of the face; the other axis are defined according to the right-hand rule (Figure 2.7). The contact point is represented in the local coordinates as $(\lambda_1^i, \lambda_2^i)$. The coordinates of the contact point p_i with respect to the object coordinate system are given by

$$\boldsymbol{P_i} = \boldsymbol{o_\lambda^i} + R_\lambda^i \begin{pmatrix} \lambda_1^i \\ \lambda_2^i \\ 0 \end{pmatrix} \tag{2.10}$$

with $\boldsymbol{o}_{\boldsymbol{\lambda}}^{\boldsymbol{i}}$ and R_{λ}^{i} denoting the origin and the rotation matrix of the local frame $\left\{\lambda_1^i, \lambda_2^i, \lambda_3^i\right\}$ with respect to the object frame, respectively.

At the grasping point, another coordinate system $\{\widehat{x}_i, \widehat{y}_i, \widehat{z}_i\}$ is introduced to represent the grasping force. The friction cone is sliced by the plane $\widehat{z}_i = 1$; the grasping force $\boldsymbol{f_i}$ in the local frame is

$$\boldsymbol{f_i} = \beta_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \tag{2.11}$$

with β_i being a nonnegative constant and $(x_i, y_i, 1)$ is the intersection point of the force $\boldsymbol{f_i}$ with the plane $\widehat{z}_i = 1$. To guarantee that the grasping force is inside the friction cone, the following must hold

$$x_i^2 + y_i^2 \leq \mu^2 \tag{2.12}$$

but with the linearized friction cone, this condition is converted in inequalities as follows

$$B_i \begin{pmatrix} x_i \\ y_i \end{pmatrix} \leq b_i \tag{2.13}$$

where B_i is a $m \times 2$ constant matrix, and b_i is a $m \times 1$ vector determined by the linearized friction cone. The grasping force $\boldsymbol{f_i}$ with respect to the object coordinate system is

$$\boldsymbol{f_i} = \beta_i R_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \tag{2.14}$$

where R_i is the rotation matrix of the local coordinate frame $\{\widehat{x}_i, \widehat{y}_i, \widehat{z}_i\}$ with respect to the object frame. The resultant wrench on the object produced by the grasping force $\boldsymbol{f_i}$ is

$$\widetilde{\boldsymbol{\omega}}_{\boldsymbol{i}} = \beta_i \begin{pmatrix} R_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \\ \boldsymbol{P_i} \times \left(R_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \right) \end{pmatrix} \tag{2.15}$$

As it is assumed that the $n-1$-fingers grasp is not FC, the convex hull $CH(\widetilde{\boldsymbol{G}})$, with $\widetilde{\boldsymbol{G}}$ the wrench matrix for the $n-1$ fingers, does not contain the origin. To find a FC grasp, the convex cone $CC(\widetilde{\boldsymbol{G}}, O)$ of the primitive contact wrenches and the origin O is built. The convex cone is the intersection of a finite set of half spaces having a common point (the cone vertex), i.e. the origin in this case. The cone is represented with

$$CC(\widetilde{\boldsymbol{G}}, O) = \left\{x \in \Re^6 | c_j^T x \geq 0,\ j = 1, 2, \ldots, t\right\} \tag{2.16}$$

where t is the number of facets of the 6-dimensional cone, and $c_j \in \Re^6$ is a constant vector. The equation $c_j^T x = 0$ represents a boundary facet of the convex cone. The necessary and sufficient condition to get a FC grasp is given in the following theorem:

Theorem: assume that the grasp of the $n-1$ fingers is not FC. The n-fingers grasp $\{P_1, P_2, \ldots, P_n\}$ is FC iff there exist proper $\beta_n \geq 0$ and (x_n, y_n) such that

1. (x_n, y_n) satisfy the condition (2.13) (i.e. the force $\boldsymbol{f_n}$ must be inside the linearized friction cone).

2. The ray $-\widetilde{\boldsymbol{\omega}}_{\boldsymbol{n}}$ intersects with the interior region of the convex cone $CC(\widetilde{\boldsymbol{G}}, O)$ (Figure 2.8)

The second condition indicates that the ray must intersect with the interior region of the convex cone, i.e. there exist (x_n, y_n) such that

$$-c_j^T \widetilde{\boldsymbol{\omega}}_{\boldsymbol{n}} > 0,\ j = 1, 2, \ldots, t \tag{2.17}$$

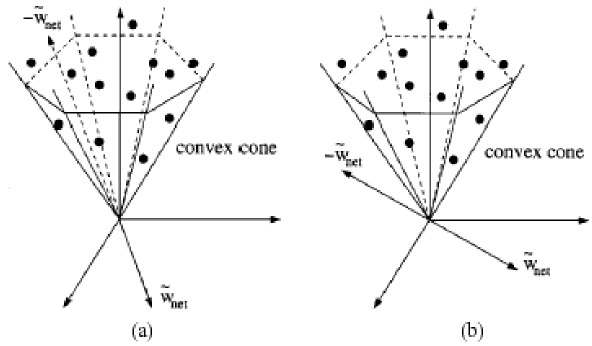

Figure 2.8: Necessary and sufficient condition to obtain a n-fingers FC grasp, a) $-\widetilde{\boldsymbol{\omega}}_{\boldsymbol{n}}$ intersects the interior of the convex cone $CC(\widetilde{\boldsymbol{G}}, O)$; b) $-\widetilde{\boldsymbol{\omega}}_{\boldsymbol{n}}$ does not intersect the interior of the cone (Source: [17]).

or, in detail,

$$c_j^T \begin{pmatrix} R_n \begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} \\ \left(\boldsymbol{o}_{\boldsymbol{\lambda}}^{\boldsymbol{n}} + R_\lambda^n \begin{pmatrix} \lambda_1^n \\ \lambda_2^n \\ 0 \end{pmatrix} \right) \times \left(R_n \begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} \right) \end{pmatrix} \leq 0, \; j = 1, 2, \ldots, t \tag{2.18}$$

In this way, the grasping point P_n that guarantees a n-fingers FC grasp is a set of parameters $(\lambda_1^n, \lambda_2^n)$ that guarantee the existence of (x_n, y_n) satisfying the inequalities (2.13) and (2.18) simultaneously. Equation (2.18) is rewritten as

$$S_n(\lambda_1^n, \lambda_2^n) \begin{pmatrix} x_n \\ y_n \end{pmatrix} \leq s_n(\lambda_1^n, \lambda_2^n) \tag{2.19}$$

where the matrix $S_n(\lambda_1^n, \lambda_2^n)$ and the vector $s_n(\lambda_1^n, \lambda_2^n)$ are linear in $(\lambda_1^n, \lambda_2^n)$. In the $\widehat{x_n} - \widehat{y_n}$ plane, (2.19) defines a convex polygon whose vertices and edges depend on the values of $(\lambda_1^n, \lambda_2^n)$. On the other hand, the restrictions (2.13) define another convex polygon whose edges and vertices are fixed. Thus, if there is a FC grasp, the two polygons must intersect each other, and the intersection region sets the parameters defining the region in face F_n such that when placing the n-th finger in this region, a FC grasp is obtained. An algorithm to find a discrete approximation to this region is presented in [16], and the complete region is found in [17], although it is computationally expensive.

An optimal location for the n-th finger can be computed, according to a particular quality measure [18]. The constraints (2.13) and (2.18) contain four variables, x_n, y_n, λ_1^n and λ_2^n, that can be grouped in a single vector z:

$$z = \begin{pmatrix} x_n \\ y_n \\ \lambda_1^n \\ \lambda_2^n \end{pmatrix} \tag{2.20}$$

The nonlinear constraints (2.18) involve the product of two of the variables. They are written in a general way as

$$gn_i(z) \leq 0, \; i = 1, \ldots, t \tag{2.21}$$

Constraints derived from the linearized friction cone are linear in the previous variables; in a general way, they are written as

$$gl_i(z) \leq 0, \; i = t, \ldots, t + m \tag{2.22}$$

Finally, if the n-th finger must be placed on the face F_n, and that face is bounded by l edges, the parameters $(\lambda_1^n, \lambda_2^n)$ must also satisfy l linear restrictions $a_j\lambda_1^n + b_j\lambda_2^n + c_j \leq 0$ ensuring that the contact point lies on the surface of the object. These constraints are

$$a_j z_3 + b_j z_4 + c_j \leq 0,\ i = t+m+1, \ldots, t+m+l \tag{2.23}$$

The quality index to find an optimal contact point is the distance L_2 between the center of mass of the object and the centroid of the contact points. This quality index is the objective function to be minimized, subject to the restrictions (2.21), (2.22) and (2.23).

2.4.2 Given the location of k fingers

The previous ideas are extended in [17] and [18] to compute the locations of $n-k$ fingers given the locations for k fingers that do not generate a FC grasp. The net wrench applied by the $n-k$ fingers is given by

$$\widetilde{\boldsymbol{\omega}}_{\boldsymbol{net}} = \sum_{i=k+1}^{n} \beta_i \begin{pmatrix} R_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \\ \boldsymbol{P_i} \times \left(R_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \right) \end{pmatrix} \tag{2.24}$$

The necessary and sufficient condition for a FC grasp is based on a theorem analogous to the one presented in the previous Subsection.

Theorem: assume that the grasp of the k fingers is not FC. The n-fingers grasp $\{P_1, P_2, \ldots, P_n\}$ is FC iff there exist proper $\beta_i \geq 0$ and (x_i, y_i) for all $i = k+1, \ldots, n$ such that

1. (x_i, y_i) satisfy the condition (2.13).
2. The ray $-\widetilde{\boldsymbol{\omega}}_{\boldsymbol{net}}$ intersects the interior of the convex cone $CC(\widetilde{\boldsymbol{G}}, O)$.

where $\widetilde{\boldsymbol{G}}$ is the convex hull of the primitive contact wrenches for the k initial fingers. This 6-dimensional convex cone can be obtained with the algorithm Qhull (available in www.qhull.org) provided that $rank(\widetilde{\boldsymbol{G}}) = 6$. However, if the primitive wrenches expand a 5-dimensional space instead of a 6-dimensional space, then the convex cone can not be obtained. This singular case often occurs when only two fingertips positions are given, or when at least two fingers are in the same face on the polyhedral object, and it is not contemplated in the algorithm.

The computation of the grasping region for each one of the $n-k$ fingers follows the ideas given in the previous Subsection. The computation of the optimal grasp uses the following vectors:

$$V = \left(\lambda_1^{k+1}, \lambda_2^{k+1}, \ldots, \lambda_1^n, \lambda_2^n\right) \tag{2.25}$$

$$z = (\beta_{k+1}x_{k+1}, \beta_{k+1}y_{k+1}, \ldots, \beta_n x_n, \beta_n y_n, V, \beta_{k+1}, \ldots, \beta_n)^T \tag{2.26}$$

The first condition for the existence of the FC grasp, i.e. (2.13), is rewritten as

$$B_i \begin{pmatrix} \beta_i x_i \\ \beta_i y_i \end{pmatrix} \leq \beta_i b_i,\ i = k+1, \ldots, n \tag{2.27}$$

thus, $(n-k) * m$ linear constraints are obtained. Condition 2 of the theorem, i.e.

$$-c_j^T \widetilde{\boldsymbol{\omega}}_{\boldsymbol{neto}} > 0,\ j = 1, 2, \ldots, t \tag{2.28}$$

denotes the FC constraint; the detailed form is similar to (2.18), which is a set of nonlinear constraints. For polyhedral objects, if the face F_i is bounded by l_i edges, the parameters $(\lambda_1^i, \lambda_2^i)$ must also satisfy l_i linear constraints, ensuring the grasping points are inside the surface:

$$f_{ij}(\lambda_1^i, \lambda_2^i) \leq 0,\ j = 1, 2, \ldots, l_i \tag{2.29}$$

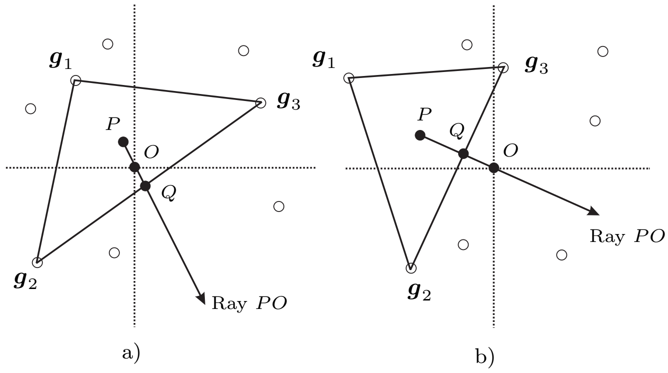

Figure 2.9: Theorem to test the FC property: a) FC grasp, b) Non FC grasp.

Equations (2.27), (2.28) and (2.29) are the restrictions for the optimization problem. The grasping parameters V that simultaneously satisfy the three conditions indicate the fingertip positions for a FC grasp. The criterion to find an optimal grasp is the minimization of the L_2 distance between the center of mass of the object, P_0, and the centroid of the polyhedron of the contact points, P_c, computed as

$$P_c = \frac{1}{n}\sum_{i=1}^{n} P_i(V) \tag{2.30}$$

The objective function is

$$\min\ u(z) = \|P_0 - P_c\|^2 = \|P_c\|^2 \tag{2.31}$$

as the origin of the object frame is located in its center of mass. Then, the search for a FC grasp is an nonlinear programming problem with a quadratic objective function subject to linear and nonlinear restrictions. It should be pointed out that the solution assumes that the faces used to place the $n-k$ fingers are selected beforehand. The proposed algorithm may also be used in the FC grasp synthesis for objects with smooth parametrized surfaces.

2.4.3 Force closure grasps for n fingers

An algorithm to compute the positions for n fingers to form a FC grasp from an initial random grasp is presented in [19]. The algorithm uses a FC test based on a ray-shooting problem [20]; the main theorem for the test is the following (Figure 2.9):

Theorem: suppose that an n-fingers frictional grasp is given, and m segments are employed to linearize each friction cone. Denote the convex hull of the primitive contact wrenches $\boldsymbol{g_i}$ by $CH(\boldsymbol{G})$. Assume that P is an interior point of $CH(\boldsymbol{G})$. The ray from the point P to the origin O of the wrench space $\Re^6$ intersects $CH(\boldsymbol{G})$ in a point Q only. A FC grasp is equivalent to that the distance d_1 between points P and Q is strictly larger than the distance d_2 between the points P and O.

To use the theorem, an interior point P of $CH(\boldsymbol{G})$ must be chosen, and then the intersection point Q of the ray PO with $CH(\boldsymbol{G})$ must be detected. A strictly positive combination of the primitive contact wrenches $\boldsymbol{g_i}$ is an interior point of the convex hull $CH(\boldsymbol{G})$, according to the following form:

$$P = \sum_{i=1}^{N} u_i \boldsymbol{g_i},\ \sum_{i=1}^{N} u_i = 1,\ u_i > 0,\ i = 1, 2, \ldots, N \tag{2.32}$$

For convenience, P is chosen as the centroid of the convex hull:

$$P = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{g_i} \tag{2.33}$$

After choosing an interior point P, the next step finds the intersection of $CH(\boldsymbol{G})$ with the ray PO. This problem is closely related to the ray-shooting problem, defined for a set M of points in $\Re^d$; it is assumed that the convex hull $CH(M)$ contains the origin, and the ray-shooting problem searches for the facet of $CH(M)$ intersected by a ray emanating from the origin. This problem can be transformed to a linear programming problem based on the duality between the convex hull and the convex polytope. However, the ray-shooting problem assumes that the convex hull contains the origin, which can not be assured in all the grasp cases. To solve this inconvenient, a coordinate translation of $-P$ on points in $\Re^6$ is applied, changing the origin to point P, which always lies inside the convex hull. After the coordinate translation, the convex hull $CH(\boldsymbol{G})$ is dual to the convex polytope

$$(\boldsymbol{g_i} - P)^T x \leq 1,\ i = 1, 2, \ldots, N \tag{2.34}$$

Let $\boldsymbol{t}$ be the direction vector of the ray PO. Based in the duality between the convex hull and the convex polytope, the ray-shooting problem is equivalent to the problem of maximizing the objective function

$$z = \boldsymbol{t}^T x \tag{2.35}$$

subject to the constraints (2.34). Suppose that the optimal solution to the previous linear programming problem is $\boldsymbol{e} = (e_1, e_2, \ldots, e_6)$. Because of the duality, the facet E of $CH(\boldsymbol{G})$ intersected by the ray PO is

$$e_1x_1 + e_2x_2 + e_3x_3 + e_4x_4 + e_5x_5 + e_6x_6 = 1 \tag{2.36}$$

Then, the intersection point Q of $CH(\boldsymbol{G})$ with the ray PO is the intersection of the hyperplane (2.36) with the ray PO.

From the qualitative test, if the distance $\|PO\|$ is larger than $\|PQ\|$, the origin is not contained in the convex hull $CH(\boldsymbol{G})$, thus the grasp is not FC. In this case, moving the primitive contact wrenches at a fixed step along directions parallel to the ray PO make the convex hull to gradually approach the origin until the origin is completely included in the hull (i.e. the fingertips on the object are gradually moved until the FC grasp is achieved). However, the primitive contact wrenches can not exactly follow the direction given by the ray PO because their force components are constant. To solve this inconvenient, a quadratic programming problem is formulated to select the fingertip positions at every iteration:

$$\min_{\boldsymbol{p} \in \Re^{3n}} \sum_{i=1}^{N} \left(\boldsymbol{g_i}^{k+1}(\boldsymbol{p}) - \boldsymbol{g_i}^{d}\right)^2 \tag{2.37}$$

subject to

$$\left(\boldsymbol{g_i}^{k+1}(\boldsymbol{p}) - \boldsymbol{g_i}^{k}\right) \bullet \vec{PO} \geq 0,\ i = 1, 2, \ldots, N \tag{2.38}$$

$$F_j(\boldsymbol{p_j}) = 0,\ j = 1, 2, \ldots, n \tag{2.39}$$

$$Norm_{jl} \bullet \boldsymbol{p_j} \geq 0,\ l = 1, 2, \ldots, t_j,\ j = 1, 2, \ldots, n \tag{2.40}$$

where $\boldsymbol{g_i}^k$ and $\boldsymbol{g_i}^{k+1}$ denote the primitive contact wrench in the k and $k+1$-th iteration; $\boldsymbol{g_i}^d$ is the primitive contact wrench at the $k+1$-th iteration, given by

$$\boldsymbol{g_i}^d = \boldsymbol{g_i}^k + \alpha \frac{\vec{PO}}{\|PO\|} \tag{2.41}$$

with α the step size (positive). $\boldsymbol{g_i}^d$ is obtained by moving $\boldsymbol{g_i}^k$ exactly along the direction parallel to the ray PO at a step size α; as this is unreachable, the objective function is chosen to minimize the distance between the actual and the desired primitive contact wrenches. The constraint (2.38) denotes that each primitive contact wrench should move along a direction differing less than 90^o with respect to PO. $\boldsymbol{p} = [p_1, \ldots, p_n]$ is the position vector and $\boldsymbol{p_j} \in \Re^3$ is associated with face j. Constraints (2.39) and (2.40) ensure that the computed position vectors $\boldsymbol{p}$ satisfy the equations of the corresponding faces and remain inside them, respectively; t_j is the edge number of the face j and $Norm_{jl}$ denotes the internal norm of the l-th boundary face of the cone formed by the origin of the object frame and the edges of face j.

The heuristic proposed to compute a FC grasp has the following steps:

1. Given the initial contact positions (randomly chosen), compute the primitive contact wrenches $\boldsymbol{g_i}$.
2. Use the test algorithm to check whether the grasp is FC. If so, the program ends with a proper solution.
3. Use equation (2.41) to compute the desired primitive contact wrenches, ${\boldsymbol{g_i}}^d$.
4. Use the quadratic programming problem defined by (2.37)-(2.40) to obtain the new position vectors and compute the new primitive contact points ${\boldsymbol{g_i}}^{k+1}$.
5. Check the distance OP^{k+1}. If it remains the same as OP^k, the program ends with no solution found; if it descends, return to step 2.

The algorithm makes that the convex hull gradually moves toward the origin until the origin is inside the hull (i.e. a FC grasp is found), or it finishes with the origin out from the convex hull while the hull remains stable when doing the iterations (a proper solution is not found), i.e. the algorithm may be trapped in local minima, which is the main inconvenient for the algorithm.

This algorithm is complemented in [21] with a procedure to calculate a good set of initial grasp points, computed with the solution of a quadratic programming problem whose objective function is the distance between the centroid of the primitive contact wrenches and the origin of the wrench space. The minimization of the objective function chooses a set of initial contacts on the object, avoiding the random selection of the previous algorithm and improving the processing time. However, the algorithm is still incomplete, i.e. the algorithm can not judge whether the problem has feasible solutions; if the algorithm fails to find a feasible solution there is no guarantee that there is none for the problem.

2.5 Equilibrium grasps with virtual fingers

Computation of equilibrium grasps for polyhedral objects using PCWF or SC fingers is presented in [22]; the computation of equilibrium grasps is closely related to FC grasp synthesis, since non-marginal equilibrium is a sufficient condition for FC grasps in the presence of friction [13]. Each finger pushes one face of the object within a convex polygonal area; each one of the fingers is replaced with virtual fingers fixed at the vertices of the region, and the equilibrium equations form a nonlinear programming problem, similar to an integer programming problem. A branch and bound algorithm is proposed to compute the solution, decomposing each finger region to search for a subregion where the moment equilibrium conditions are satisfied (the nonlinearity in the problem arises from this condition).

Table 2.1: Main references in grasp synthesis for polyhedral objects.

Reference	No. fingers	Generated grasp
[10], 1988	2 SC	Maximal independent contact regions
[11], 1994	3 PCWF	Optimal grasp with the criterion of decoupled forces
[12], 1993	3 and 4 PCWF	
	Concurrent grasps	Maximal independent contact regions
[13], 1997	4 PCWF	
	Concurrent grasp	Maximal independent contact regions
[14], 1995	4 PCWF	
	Concurrent and flat pencil grasps	Maximal independent contact regions
	Regulus grasp	Generation of one grasp
[16], 1999	n PCWF given the position of $n-1$ fingers	Feasible grasp region for the n-th finger
[17], 2001	n FPC or PCWF given the position of k fingers, $k \geq 3$	Feasible grasp region for the $n-k$ fingers, and optimal grasp minimizing the distance between the centroid of the grasp polyhedron and CM
[18], 2000	n FPC or PCWF, given the position of k fingers, $k \geq 3$	Optimal grasp minimizing the distance between the centroid of the grasp polyhedron and CM
[19], 2000	n FPC or PCWF	One FC grasp
[21], 2001	n FPC or PCWF	One FC grasp

Chapter 3

Grasp synthesis for complex objects

Grasp synthesis in 3D objects initially considered only polyhedral objects; the proposed algorithms are efficient when the number of faces of the object is low. However, new techniques are required when dealing with 3D complex objects (i.e. non-polyhedral objects and even polyhedral objects with a large number of faces), because of the large amount of information required to model the object, usually obtained from a CAD system or through sensors such as laser scanners or video cameras. The surface of the object is usually modeled with a cloud of discrete points, a triangular mesh or smooth curved surfaces.

3.1 Grasp synthesis for objects described with continuous surfaces

3.1.1 Two-finger grasps

In 3D smooth convex objects there always exists a FC grasp with two SC fingers in antipodal points, or with three fingers if no two of the contact points are antipodal [23]. An object is considered to be smooth if the boundary ∂A is diffeomorphic to the sphere S^2 (it does not have holes). Two points are antipodal if the ray trough each point in the direction of the inward normal contains the other point, i.e. their normals are collinear and in opposite direction. No algorithm to compute two or three finger grasps is proposed in [23].

The grasp synthesis for arbitrary smooth (convex and not convex) objects with two SC fingers is considered in [24]. Convex objects can be grasped by squeezing grasps only; non convex objects can be grasped by squeezing and possibly expanding grasps. The surface for a smooth object can be described by a 1 to 1 parametric function:

$$\boldsymbol{s}(\boldsymbol{u}) = [x(\boldsymbol{u}), y(\boldsymbol{u}), z(\boldsymbol{u})]^T \, , \; \boldsymbol{u} \in S^2 \tag{3.1}$$

where $\boldsymbol{u} \in S^2$ is called a contact variable. Since the function is 1 to 1, $\boldsymbol{u_0}$ can be used instead of $\boldsymbol{s}(\boldsymbol{u_0})$ to represent a contact point on the object. A contact configuration of a 2-fingers grasp is defined as $\boldsymbol{q} = (\boldsymbol{u_1}, \boldsymbol{u_2})$, with $\boldsymbol{u_1} \neq \boldsymbol{u_2}$ and $\boldsymbol{u_1}, \boldsymbol{u_2} \in S^2$. If $\Gamma = \left\{(\boldsymbol{u_1}, \boldsymbol{u_2}) | \boldsymbol{u_1} = \boldsymbol{u_2} \in S^2\right\}$ denotes the set of physically unrealizable contact configurations (as it implies that the two fingers would be on the same point), the contact configuration space D is

$$D = S^2 \times S^2 \setminus \Gamma \tag{3.2}$$

According to [10], a 2-fingers grasp $\boldsymbol{q} = (\boldsymbol{u_1}, \boldsymbol{u_2})$ is FC iff the line connecting the points $\boldsymbol{s}(\boldsymbol{u_1})$ and $\boldsymbol{s}(\boldsymbol{u_2})$ lies strictly inside both internal or external friction cones for squeezing or expanding grasps, respectively. This condition is verified by antipodal points, i.e. points that fulfill the following conditions:

$$[\boldsymbol{s}(\boldsymbol{u_1}) - \boldsymbol{s}(\boldsymbol{u_2})] \cdot \boldsymbol{t}(\boldsymbol{u_1}) = 0 \tag{3.3}$$

$$[\boldsymbol{s}(\boldsymbol{u_2}) - \boldsymbol{s}(\boldsymbol{u_1})] \cdot \boldsymbol{t}(\boldsymbol{u_2}) = 0 \tag{3.4}$$

$$\boldsymbol{n}(\boldsymbol{u_1}) + \boldsymbol{n}(\boldsymbol{u_2}) = 0 \tag{3.5}$$

where $\boldsymbol{n}(\boldsymbol{u_1})$ and $\boldsymbol{n}(\boldsymbol{u_2})$ are the unit outward pointing normal vectors, and $\boldsymbol{t}(\boldsymbol{u_1})$ and $\boldsymbol{t}(\boldsymbol{u_2})$ are unitary tangent vectors in the tangent spaces at $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$.

A FC squeezing grasp satisfies:

$$\boldsymbol{n}(\boldsymbol{u_1}) \cdot \frac{\boldsymbol{s}(\boldsymbol{u_1}) - \boldsymbol{s}(\boldsymbol{u_2})}{\|\boldsymbol{s}(\boldsymbol{u_1}) - \boldsymbol{s}(\boldsymbol{u_2})\|} > \cos\theta \tag{3.6}$$

$$\boldsymbol{n}(\boldsymbol{u_2}) \cdot \frac{\boldsymbol{s}(\boldsymbol{u_2}) - \boldsymbol{s}(\boldsymbol{u_1})}{\|\boldsymbol{s}(\boldsymbol{u_2}) - \boldsymbol{s}(\boldsymbol{u_1})\|} > \cos\theta \tag{3.7}$$

where $\tan\theta = \mu$, with μ the friction coefficient. The conditions for an expanding grasp are

$$\boldsymbol{n}(\boldsymbol{u_1}) \cdot \frac{\boldsymbol{s}(\boldsymbol{u_1}) - \boldsymbol{s}(\boldsymbol{u_2})}{\|\boldsymbol{s}(\boldsymbol{u_1}) - \boldsymbol{s}(\boldsymbol{u_2})\|} < -\cos\theta \tag{3.8}$$

$$\boldsymbol{n}(\boldsymbol{u_2}) \cdot \frac{\boldsymbol{s}(\boldsymbol{u_2}) - \boldsymbol{s}(\boldsymbol{u_1})}{\|\boldsymbol{s}(\boldsymbol{u_2}) - \boldsymbol{s}(\boldsymbol{u_1})\|} < -\cos\theta \tag{3.9}$$

Inequalities (3.6)-(3.9) define the feasible grasping regions in D; in these regions the grasps are FC. A "grasping energy function" $E : D \to \Re$, depending on the distance between the points, is defined as

$$E(\boldsymbol{u_1}, \boldsymbol{u_2}) = \frac{1}{2}\kappa \|\boldsymbol{s}(\boldsymbol{u_1}) - \boldsymbol{s}(\boldsymbol{u_2})\|^2 \tag{3.10}$$

A pair of antipodal points $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ for a 3D object correspond to critical points of E in the feasible grasp region. However, the optimization of E is difficult since D does not admit a global parametrization. To overcome this inconvenient, 3D objects are modeled with spherical product surfaces, a technique used to represent 3D object surfaces taking as a base 2D curves, that may be parametrized trigonometric curves or B-splines. For instance, the generic superellipsoids (a particular superquadric surface) are defined with the following spherical product:

$$s(u,v) = \begin{pmatrix} \cos^{\epsilon_1} u \\ a_3 \sin^{\epsilon_1} u \end{pmatrix} \otimes \begin{pmatrix} a_1 \cos^{\epsilon_2} v \\ a_2 \sin^{\epsilon_2} v \end{pmatrix} = \begin{pmatrix} a_1 \cos^{\epsilon_1} u \cos^{\epsilon_2} v \\ a_2 \cos^{\epsilon_1} u \sin^{\epsilon_2} v \\ a_3 \sin^{\epsilon_1} u \end{pmatrix} \tag{3.11}$$

with $-\pi/2 \leq u \leq \pi/2,\ -\pi \leq v \leq \pi$; equation (3.11) considers five parameters, defining the shape and size of the superellipsoid; ϵ_1 and ϵ_2 are the deformation parameters that control the shape of the primitive, and a_1, a_2 and a_3 define the primitive shape in directions x, y, and z respectively [25]. The 3D spherical product surface is defined as

$$s(u,v) = f(u) \otimes g(v) - [f_1(u)g_2(v), f_1(u)g_2(v), f_2(u)] \tag{3.12}$$

where $f(u)$ and $g(v)$ are 2D curves:

$$f(u) = [f_1(u), f_2(u)]\,,\ u \in I_u = [u_o, u_1] \tag{3.13}$$

$$g(v) = [g_1(v), g_2(v)]\,,\ v \in I_v = [v_o, v_1] \tag{3.14}$$

with $f(u)$ and $g(v)$ parametric trigonometric curves. This definition can be extended to represent a richer set of objects using B-splines; let $f : I_u = I_m \to \Re^2$ be an open cubic B-spline curve of m segments and $g : I_v = I_n \to \Re^2$ a closed cubic B-spline curve of n segments. To guarantee object surface smoothness, $f(u)$ and $g(v)$ must satisfy the following conditions:

- $f(u)$ and $g(v)$ must be regular curves
- The curve $f(u)$ intersects the y-axis at $f(0)$ and $f(m)$ only; the tangents $f'(0)$ and $f'(m)$ must have zero slope.

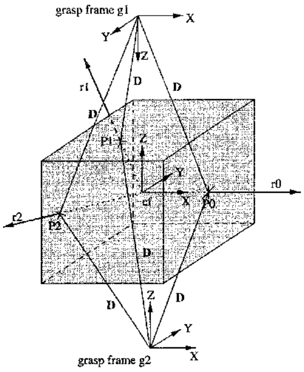

Figure 3.1: Heuristic generation of three-fingers grasps (Source: [26]).

- The tangent vector and the position vector of a point on $g(v)$ are not parallel, i.e. $g(v) \neq \alpha g'(v)$ for some $\alpha \neq 0$, or $g_1 g_2' - g_1' g_2 \neq 0$.

These surfaces can be be deformed to approximate the real shape of an object with transformations such as linear stretching, tapering or bending. The spherical product mapping $I_m \times I_n$ onto a surface diffeomorphic to S^2 is not a 1 to 1 mapping because the two polar points, $\boldsymbol{p_n} = \boldsymbol{s}(0, v)$ and $\boldsymbol{p_s} = \boldsymbol{s}(m, v)$ with $v \in I_n$ are degenerated curves; however, the normal vectors at the polar points are well defined and they are continuous in their neighborhoods.

The domain of the spherical product surface is $I_u \times I_v$, so the contact configuration space is $D^* = I_u \times I_v \times I_u \times I_v$, which is topologically different from D. However, this parametrization allows the optimization of the energy function (3.10) on the feasible grasp region D^*, thus allowing the computation of the antipodal points defining the FC grasp on the object.

3.1.2 Three, four and five-fingers grasps

Heuristic grasp generation has been proposed to synthesize grasps with more than two contact fingers. For instance, a simple heuristic search to synthesize FC grasps with three fingers is presented in [26]. A coordinate system with arbitrary origin and orientation is generated inside the bounding box of the object (Figure 3.1). Three rays are generated in predefined directions from the origin of this frame; the first one is emitted along the x axis, the second and third along the x axis rotated about the z axis by 120° and 240°, respectively. If all the three rays yield one penetration point, the set of intersection points with the object surface is a grasp candidate. For non-convex objects, more than one penetration point penetrating the object surface from the inside can be obtained; in this case, one of the penetration points is chosen randomly. This heuristic search does not take into account the friction coefficient, but the generated grasp is FC according to conditions presented in Section 2.2.

Grasp candidates are filtered to exclude candidates which can not lead to feasible grasps. The first filter discards candidates with grasp points too close to supporting surfaces; the second filter verifies that the grasping points are not further apart than the hand can spread its fingers. The third filter avoids collisions between the hand and the environment; the implementation generates two points at a fixed distance D measured from the contact points, the hand frame is placed in these two origins, and a collision check between the hand and the environment is performed. Candidate grasps passing the previous filters are ordered according to a quality measure, and the algorithm finally chooses the best quality grasp within the initial candidate grasps.

The previous work is extended in [27] to four-fingers grasps. The synthesis procedure follows the same ideas: a set of candidate grasps is generated and filtered using the same

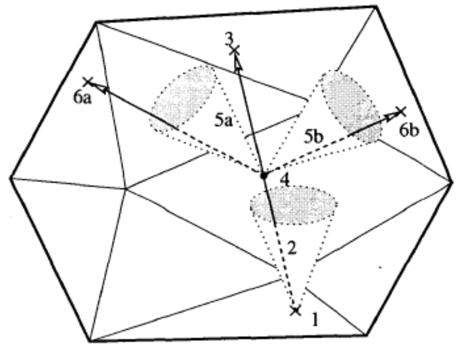

Figure 3.2: Heuristic generation of four-fingers grasps (Source: [27]).

criteria, and the candidates overcoming the filters are ordered according to a quality measure to finally choose the best candidate grasp among the set of initial candidates. The generation of candidate grasps is illustrated in Figure 3.2. A point on the object surface is arbitrarily chosen as the first contact point (1). A ray direction (2) deviated from the surface normal by an arbitrary angle (smaller than the angle of the friction cone) is computed; a penetration point of this ray is chosen as the second contact point (3). Next, the center point (4) of the two already existing grasp points is calculated, and two rays (5a, 5b) are emitted in two directions which are arbitrary chosen with a distribution between the following two distributions:

- The rays (2), (5a) and (5b) lie on a plane, separated by an angle of 120º. This criterion is optimal for gripper kinematics.
- (5a) and (5b) are perpendicular to (2), and the separation angle between (5a) and (5b) is 120º. This criterion emulates the optimal grasp points location for a tetrahedron.

The penetration points (6a) and (6b) are the two missing contact points. In this way, concurrent four-fingers grasps fulfilling the conditions presented in Section 2.3 are obtained.

3.1.3 Grasps with any number of fingers

Several generic algorithms have been presented, i.e. algorithms to synthesize grasps for any 3D object with smooth curved surfaces and with any number of contacts. An algorithm to synthesize grasps based on the concept of the Q distance (or Q norm) is presented in [28]. The Q norm ($\|\cdot\|_Q$), or gauge function, is a grasp quality measure defined for a convex compact set $Q \subset \Re^m$ which contains the origin of the reference system in $\Re^m$:

$$g_Q\left(\boldsymbol{a} \in \Re^m\right) = \inf_{\boldsymbol{a} \in \gamma Q, \gamma > 0} \gamma \tag{3.15}$$

The Q distance is the minimum scale factor required for the set Q to contain a given point $\boldsymbol{a}$, i.e. it quantifies the maximum wrench that can be resisted in a predefined set of directions, given by the set Q. A sphere in terms of $\|\cdot\|_Q$ (a $\|\cdot\|_Q$ sphere) centered in the origin is defined as $S_Q(\rho) = \left\{\boldsymbol{a} | \left\|\boldsymbol{a}\right\|_q \leq \rho\right\}$. To implement the grasp planning algorithms, Q is taken as a polyhedral set. Let $\boldsymbol{p} \in \Re^m$ and $A \subset \Re^m$ be a point and a convex polyhedron, respectively; the Q^+ distance is defined as

$$d_Q^+\left(\boldsymbol{p}, A\right) = \min_{\boldsymbol{a} \in A} \|\boldsymbol{a} - \boldsymbol{p}\|_Q \tag{3.16}$$

This distance can be interpreted as the radius of the smallest $\|\cdot\|_Q$ sphere in contact with $A - \{\boldsymbol{p}\}$.

A grasp with primitive contact wrench matrix $\boldsymbol{G}$ is FC if $d_Q^+(0, CH(\boldsymbol{G})) = 0$. However, this is a necessary but not sufficient FC condition. This makes necessary the definition of the distance d_Q^-, as

$$d_Q^-(\boldsymbol{p}, A) = -\min_{\boldsymbol{a}\in\partial A} \|\boldsymbol{a} - \boldsymbol{p}\|_Q \tag{3.17}$$

where ∂A denotes the boundary set of A. By definition, $d_Q^-(\boldsymbol{p}, A) \leq 0$ for any $\boldsymbol{p} \in A$. The distance $d_Q^-(\boldsymbol{p}, A)$ has a geometric interpretation: it is the radius of the largest $\|\cdot\|_Q$ sphere contained in $A - \{\boldsymbol{p}\}$. This geometric interpretation is similar to the classical quality measure described in [29, 30], where the grasp quality is measured with the L_2 norm of the largest wrench that the robotic hand can produce on the grasped object in the worst possible direction; in this case, the L_2 norm is replaced with the $\|\cdot\|_Q$ norm, and a similar measure is obtained.

Assuming that the grasped object is piecewise smooth, and each finger is placed at one of the smooth surfaces, it can be stated that the primitive contact wrenches depend on the grasp configuration $\boldsymbol{u}$, i.e. $\boldsymbol{G} = (\boldsymbol{g_{11}}(\boldsymbol{u}), \ldots, \boldsymbol{g_{nm}}(\boldsymbol{u}))$. Therefore, if the definition of d_Q^- is used with $\boldsymbol{p} = 0$, $A = CH(\boldsymbol{G}(\boldsymbol{u}))$, the following sufficient and necessary condition is obtained:

Theorem: a grasp is FC iff $d_Q^-(\boldsymbol{u}) < 0$.

Therefore, d_Q^- provides a qualitative test of the FC property and quantifies the capability of the grasp to resist external loads and/or disturbances, and becomes useful to obtain and optimize the configuration of a FC grasp. Distances d_Q^+ and d_Q^- are computed efficiently solving linear programming problems (with the simplex method), and their derivatives are easily computed in most of the cases; these properties make easier the implementation of a grasp planning algorithm. The Q distance can be defined in general as

$$d_Q(\boldsymbol{u}) = \begin{cases} d_Q^+(\boldsymbol{u}), \ 0 \notin CH(\boldsymbol{G}(\boldsymbol{u})) \\ d_Q^-(\boldsymbol{u}), \ 0 \in CH(\boldsymbol{G}(\boldsymbol{u})) \end{cases} \tag{3.18}$$

The grasp planning algorithm must minimize $d_Q(\boldsymbol{u})$ in the grasp configuration space. The constraints for the minimization problem are obtained by restricting the contacts to lie on some smooth pieces of the surface. These constraints are denoted in general as $C_i(\boldsymbol{u}) \geq 0$, $i = 1, \ldots, \kappa$, and it is assumed that the $C_i(\boldsymbol{u})$'s are smooth functions. To handle the constraints, a potential field is introduced; let $\epsilon > 0$ be a threshold which specifies the active region of the potential field, and $\Im_\epsilon(\boldsymbol{u}) = \{i | C_i(\boldsymbol{u}) < \epsilon\}$. The potential field, for $\gamma > 0$, is defined by

$$P(\boldsymbol{u}) = \sum_{i\in\Im_\epsilon(\boldsymbol{u})} \left(\frac{1}{C_i(\boldsymbol{u})} - \frac{1}{\epsilon}\right)^\gamma \tag{3.19}$$

Using this potential function, the grasp planning problem reduces to minimize the following unconstrained objective function:

$$f(\boldsymbol{u}) = d_Q(\boldsymbol{u}) + \nu P(\boldsymbol{u}) \tag{3.20}$$

with ν a weighting coefficient, usually selected as a small positive number. The minimization of the objective function (3.20) is achieved with a descent search algorithm, taking advantage of the differentiability properties of $d_Q(\boldsymbol{u})$. The initial value for the problem is a grasp $\boldsymbol{u_0}$ fulfilling the restrictions $C_i(\boldsymbol{u_0}) \geq 0$, $i = 1, \ldots, \kappa$. The proposed algorithm can be used for grasp planning with any number of contacts on 3D objects with smooth curved surfaces. It is also used to synthesize grasps when k of the n contact points are fixed and the others are to be determined to get a FC grasp.

The gauge function is generalized in [31, 32] to create a new numerical FC test; the detailed development of the test is found in [33]. A grasp with matrix of primitive contact wrenches $\boldsymbol{G}$ is FC iff $0 \in int(CH(\boldsymbol{G}))$, or equivalently

1. $0 \in ri(CH(\boldsymbol{G}))$
2. $rank(\boldsymbol{G}) = 6$

where $ri(CH(\boldsymbol{G}))$ denotes the relative interior of $CH(\boldsymbol{G})$. The second condition is easily verified with a singular value decomposition of $\boldsymbol{G}$. For the first condition, a necessary and sufficient condition is that there exists a point $\boldsymbol{w_0} \in ri(CH(\boldsymbol{G}))$ and a nonnegative number $\alpha < 1$ such that $\boldsymbol{w} \in \{\boldsymbol{w_0}\} + \alpha\,(CH(\boldsymbol{G}) - \boldsymbol{w_0})$.

To implement the numerical test, the average wrench $\boldsymbol{w_0}$ is used ($\boldsymbol{w_0} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{g_i}$, with $N = nm$), as $\boldsymbol{w_0} \in ri(CH(\boldsymbol{G}))$. The numerical test is

$$\psi(\boldsymbol{G}) = \min_{0 \in \{\boldsymbol{w_0}\} + \alpha(CH(\boldsymbol{G}) - \boldsymbol{w_0}),\ \alpha \geq 0} \alpha \tag{3.21}$$

$\psi(\boldsymbol{G})$ provides a measure of how much $CH(\boldsymbol{G})$ must be grown outward from $\boldsymbol{w_0}$ to contain the origin of the wrench space. In particular, if $rank(\boldsymbol{G}) = 6$ then $\psi(\boldsymbol{G})$ is the gauge function of $CH(\boldsymbol{G})$. As a property of the gauge function, $0 \in int(CH(\boldsymbol{G}))$ if and only if $\alpha < 1$. Therefore, α provides a numerical test for the first FC condition, so $\alpha < 1$ indicates that $0 \in ri(CH(\boldsymbol{G}))$. The measure is computed with the following linear programming problem:

$$\psi(\boldsymbol{G}) = \min \sum_{i=1}^{N} \alpha_i \tag{3.22}$$

$$subject\ to \begin{cases} \boldsymbol{w_0} + \sum_{i=1}^{N} \alpha_i\,(\boldsymbol{g_i} - \boldsymbol{w_0}) = 0 \\ \alpha_i \geq 0 \end{cases} \tag{3.23}$$

The grasp configuration is given by a vector $\boldsymbol{u}$ with parameters specifying the position of the contact points. It is assumed that the object has a piecewise smooth surface, and each contact point is positioned at one of the smooth pieces. Thus, the primitive wrenches can be represented as smooth functions of the grasp configuration, i.e. $\boldsymbol{G}(\boldsymbol{u}) = (\boldsymbol{g_{11}}(\boldsymbol{u}), \ldots, \boldsymbol{g_{nm}}(\boldsymbol{u}))$; the corresponding numerical test is $\psi(\boldsymbol{u})$. Since $\psi(\boldsymbol{u}) < 1$ is the necessary condition for the FC property, and the value of $\psi(\boldsymbol{u})$ provides a measure on how far the grasp is from losing the closure property, a natural way to compute a FC grasp is to minimize $\psi(\boldsymbol{u})$. This optimization problem can be solved with a descent search algorithm.

A necessary but not sufficient condition for the FC is $\psi(\boldsymbol{u}) < 1$; however, the solution must also verify that $rank(\boldsymbol{G}) = 6$. This condition is equivalent to $\det(\Phi(\boldsymbol{u})) > 0$, with $\Phi(\boldsymbol{u}) = \boldsymbol{G}(\boldsymbol{u})\boldsymbol{G}^T(\boldsymbol{u})$. A larger value of $\det \Phi(\boldsymbol{u})$ suggest a better ability to resist external perturbation wrenches. Thus, in order to plan FC grasps with a desirable performance, the following objective function must be minimized

$$f(\boldsymbol{u}) = \psi^{\alpha}(\boldsymbol{u}) - \eta g\,(\phi(\boldsymbol{u})) \tag{3.24}$$

where $\phi(\boldsymbol{u}) = \det \Phi(\boldsymbol{u})$, and $g(\cdot)$ is a monotonous function; for instance, $g\,(\phi(\boldsymbol{u})) = \phi^{\beta}(\boldsymbol{u})$, or $g\,(\phi(\boldsymbol{u})) = \ln \phi(\boldsymbol{u})$. The proposed algorithm has a general application, and is computationally more efficient that the algorithm proposed in [28].

3.1.4 Randomized grasps

Most of the algorithms to synthesize grasps on polyhedral or continuous surface objects generate grasps assuming that the faces to be grasped are previously chosen; however, few works have included the problem of appropriate face selection. Moreover, a widely used technique to represent an arbitrary object is approximate its surface with a large triangular mesh (specially when the object information comes from cameras or laser scanners; a common number of faces is $10^3 - 10^5$); since the computational cost of the algorithms depends on the number of faces describing the object, the previously presented algorithms are expected to take long time to generate a good grasp. However, a random generation of grasps may be quickly and appropriate for on-line grasp synthesis (in real applications), computing very fast and good, although not optimal, grasps [34].

The algorithm in [34] generates grasp candidates randomly choosing contacts on the object surface. The candidates are pre-filtered to discard those candidates for which a computed external force f_{ext} breaks the grasp; this happens if the angles α_i between the force vector f_{ext} and the normals of all the contacts i is bigger than $90^o + \arctan(\mu)$, where μ is the friction coefficient. Two external forces are tested, computed as follows: (1) Choosing the two contact normals with the largest angle between them, f_{ext} is in the opposite direction to the bisector between them, (2) f_{ext} is in the opposite direction to the average of all the contact normals. Among the candidates passing the pre-filtering stage, the FC grasp candidates are selected. The quality of the FC grasp candidates is measured according to the classical quality index presented in [30], which quantifies the maximum wrench that the grasp can resist with independence of the wrench direction; the algorithm finally chooses the best quality grasp. The algorithm is tested for 3, 4 and 5 fingers, and it is stated that the generated grasps have a quality similar to the expected quality from a human grasp on the same objects. The calculation complexity of the random grasp planner depends only on the object form, not on the number of faces composing the object surface.

3.2 Grasp synthesis for discretized objects

The previous Section has described several algorithms to synthesize grasps in objects with piecewise smooth surfaces; however, to apply these algorithms to real objects, the surface of the object must be parametrized, which usually is not an easy task. Another common way to represent complex objects is by using triangular meshes; the application of the grasp synthesis algorithms for polyhedral objects to objects described with triangular meshes has a high computational cost, as it was previously described. To overcome this problem, the surface of the 3D object is sampled, generating a set of surface points, whose position and corresponding normal direction is stored. This approach allows the application of grasp synthesis algorithms to arbitrary objects, assuming the number of points in the cloud of points is enough to accurately describe the object surface. A first approach to synthesize FC grasps on a set of discrete surface points is to exhaustively search for a set of points fulfilling the FC condition using all the possible combinations of points. Evidently, the complexity of such algorithm is very high: $O(N^n)$, with N the number of points on the object surface and n the number of contacts. To reduce this complexity, several algorithms with oriented searches have been proposed.

3.2.1 A complete algorithm

An algorithm to synthesize FC grasps with 7 frictionless contacts is proposed in [35]. The grasped object has been discretized previously, so a large cloud Ω of points $\boldsymbol{p_i}$ as well as their normals $\boldsymbol{n_i}$ is available. Then, a large collection of contact wrenches $\boldsymbol{g_i}$ can be obtained:

$$\boldsymbol{g_i} = \begin{pmatrix} \boldsymbol{n_i} \\ \boldsymbol{p_i} \times \boldsymbol{n_i} \end{pmatrix} \tag{3.25}$$

The algorithm starts with an initial set of seven contacts randomly chosen among the set of points; if the selected grasp is FC, the algorithm finishes; otherwise, the initial contacts are iteratively exchanged with other candidate locations until a FC grasp is obtained. The FC verification uses the ray-shooting test [20], presented in Subsection 2.4.3.

The exchange procedure has two phases. First, the set of points Ω is divided into two sets according to the relative locations of their wrenches with respect to a separating hyperplane (Figure 3.3). The hyperplane E that contains the facet of $CH(\boldsymbol{G})$ intersected by the ray PO between the centroid P of $CH(\boldsymbol{G})$ and the origin O of the wrench space has as generic equation

$$\sum_{i=1}^{6} e_i x_i = 1 \tag{3.26}$$

The separating hyperplane Y is a hyperplane parallel to E and passing through the origin;

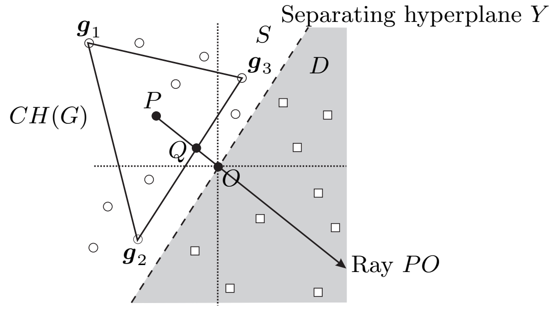

Figure 3.3: Sets of wrenches for a FC grasp with frictionless fingers (in a hypothetical 2D wrench space).

its equation is

$$\sum_{i=1}^{6} e_i x_i = 0 \tag{3.27}$$

and it divides the wrench space in two subspaces Y^+ and Y^-, where Y^+ is the half space $e_1x_1+e_2x_2+e_3x_3+e_4x_4+e_5x_5+e_6x_6 \geq 0$ and Y^- is the half space $e_1x_1+e_2x_2+e_3x_3+e_4x_4+e_5x_5+e_6x_6 < 0$. The wrenches are classified in two sets: S, containing the wrenches located on the same side of the hyperplane as the 7 initial wrenches, and D, with the wrenches lying on the other side.

The second phase iteratively replaces one of the initial wrenches with a wrench randomly selected from the set D. Let $\boldsymbol{g_j}$ be the initial wrenches; the centroid P of these wrenches is

$$P = \frac{1}{7}\sum_{j=1}^{7} \boldsymbol{g_j} \tag{3.28}$$

When an initial wrench $\boldsymbol{g_j}$ is replaced with $\boldsymbol{g_k}$, the new centroid will be

$$P_{jk} = P - \boldsymbol{g_j}/7 + \boldsymbol{g_k}/7,\ j = 1, 2, \ldots, 7 \tag{3.29}$$

The algorithm chooses the wrench j to be replaced when the maximum approach from P to O is achieved:

$$j = i \mid \min_i \|P_{ik}O\|,\ i = 1, 2, \ldots, 7 \tag{3.30}$$

It usually happens that the minimum distance $\|PO\|$ with the new wrench $\boldsymbol{g_k}$ is larger than in the previous iteration (the convex hull moves farther from the origin, instead of approaching it); in this case another wrench in the set D is chosen until $\|PO\|$ diminishes with respect to the previous iteration. If all the wrenches in D are tried and the minimum distance $\|PO\|$ can not be diminished, the algorithm is in a deadlock. To escape, the algorithm picks up 7 new initial wrenches and restarts the process until the convex hull contains the origin, i.e. until a FC grasp is obtained.

The previous heuristic algorithm is extended in [36] for any number of contacts with or without friction. The algorithm starts with a random grasp selected from the set of points Ω. If the grasp is not FC, the algorithm iteratively moves the grasp points to shorten the distance between the convex hull and the origin of the wrench space. This algorithm, unlike the previous one in [35], considers that each point is connected with 4 neighboring points, and all the possible moves for each finger are studied in each iteration, looking for the grasp position that achieves the minimum distance $\|PO\|$. The algorithm iterates until it finds a local minimum; when this happens, the separating hyperplane divides the total set of points Ω in three subsets to ease the search of the FC grasp, by discarding points that do not contribute to the solution. The subsets are:

$$\Omega(Y^-) = \left\{\boldsymbol{p_i} \in \Omega | \boldsymbol{g_{ij}} \in Y^-,\ for\ all\ j = 1, 2, \ldots, m\right\} \tag{3.31}$$

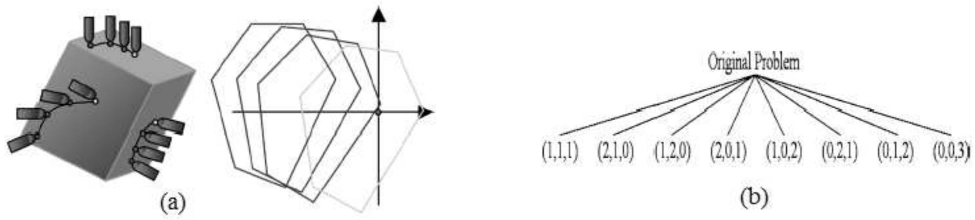

Figure 3.4: Grasp of discretized objects with n fingers: a) Local heuristic search b) Search tree to decompose the problem with 3 PCWF fingers (Source: [36]).

$$\Omega(Y^+) = \{\boldsymbol{p_i} \in \Omega | \boldsymbol{g_{ij}} \in Y^+, \ for \ all \ j = 1, 2, \ldots, m\} \tag{3.32}$$

$$\Omega(Y^0) = \{\boldsymbol{p_i} \in \Omega | \boldsymbol{p_i} \notin \Omega(Y^-), \ \boldsymbol{p_i} \notin \Omega(Y^+)\} \tag{3.33}$$

where $\boldsymbol{g_{ij}}$ denotes the primitive contact wrenches (m for each finger); $\Omega(Y^-)$ and $\Omega(Y^+)$ are the subsets of Ω with contact points whose m wrenches are all located on the half spaces Y^- and Y^+, respectively, and $\Omega(Y^0)$ is the subset of contact points whose m wrenches are distributed to both sides of the separating hyperplane. Note that in a grasp with FPC fingers the subset $\Omega(Y^0)$ does not exist, since each contact point has only one corresponding primitive contact wrench; this simplifies the grasp search.

In a FC grasp, the n grasp points should not all be in subset $\Omega(Y^+)$ or subset $\Omega(Y^-)$. This suggests the division of the problem in the original set into problems in the subsets, based on the existence conditions for a FC grasp. The grasp points should be selected from the subsets according to the following combinations:

- $\Omega(Y^-)$, $\Omega(Y^+)$ and $\Omega(Y^0)$
- $\Omega(Y^-)$ and $\Omega(Y^+)$ only
- $\Omega(Y^-)$ and $\Omega(Y^0)$ only
- $\Omega(Y^+)$ and $\Omega(Y^0)$ only
- $\Omega(Y^0)$ only

For instance, for a 3-fingers PCWF grasp, eight problems in the subsets must be considered, which are grasps consisting of

- one grasp point from $\Omega(Y^-)$, one from $\Omega(Y^+)$ and one from $\Omega(Y^0)$.
- two grasp points from $\Omega(Y^-)$ and one from $\Omega(Y^+)$.
- one grasp point from $\Omega(Y^-)$ and two from $\Omega(Y^+)$.
- two grasp points from $\Omega(Y^-)$ and one from $\Omega(Y^0)$.
- one grasp point from $\Omega(Y^-)$ and two from $\Omega(Y^0)$.
- two grasp points from $\Omega(Y^+)$ and one from $\Omega(Y^0)$.
- one grasp point from $\Omega(Y^+)$ and two from $\Omega(Y^0)$.
- all three grasp points from $\Omega(Y^0)$.

The subproblems obtained are represented as child nodes in a search tree whose root is the original problem (Figure 3.4). The convention $\langle i, j, k \rangle$ in the node denotes that i, j and k

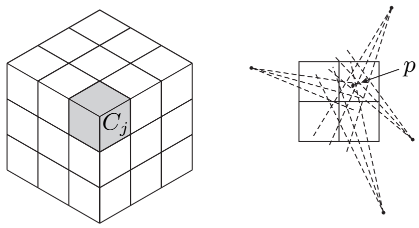

Figure 3.5: Concurrent grasps for discretized objects: a) Cube discretized in voxels, b) Set M of 4 points whose friction cones contain the point p.

grasp points are selected from subsets $\Omega(Y^-)$, $\Omega(Y^+)$ and $\Omega(Y^0)$, respectively. To determine which child node should be traversed first, the following heuristic function is defined:

$$f(\langle a_1, a_2, \ldots, a_t \rangle) = \sum_{i=1}^{t} \left(\frac{7}{t} - a_i \right)^2 \tag{3.34}$$

where t is the number of subsets corresponding to the node and a_i denotes the number of contact points in subset i. For instance, for the right-most node in Figure 3.4, $t = 3$, $a_1 = 0$, $a_2 = 0$ and $a_3 = 3$. The heuristic favors the nodes with eligible contact points more evenly distributed over the point subsets, considering they have a higher chance to contain a FC grasp; therefore, a node with a smaller value of f will be traversed before other node with a higher value. When one node is selected, a local search is performed in the corresponding subset. If the distance between the convex hull and the origin has a new local minimum, the corresponding separating hyperplane subdivides the problem using again the existence conditions of the FC grasp. The search is performed recursively or until all the nodes have been explored.

The complexity of the search tree does not depend on the total number of points representing the object, but on the number n of fingers and on the number of local minima in the grasp configuration space (i.e. by the discrete geometry of the object surface). The number of nodes in the search tree goes down with the increasing number of fingers, because the chance of finding a solution with the initial grasp increases as n increases. The proposed algorithm is complete: it finds a solution if one exists, otherwise it reports that there is no solution. However, the failure to find a solution in the discrete domain does not imply that no FC grasp exists on the continuous surface of the object. The grasp obtained with this algorithm does not assure any optimality.

3.2.2 Concurrent grasps

A heuristic algorithm to search for 4-fingers concurrent FC grasps is presented in [37]. The objective is to generate a number of 4-fingers concurrent FC grasps to provide the user with a large set of grasps, so the user can choose an optimum one according to a quality measure appropriate for the particular task, for instance, in regrasp planning. The FC test is based on the conditions presented in Subsection 2.3.1, i.e. P1) there exist four lines in the corresponding double-sided friction cones that intersect in a single point, and P2) the vectors parallel to these lines and lying in the internal friction cones at the contact points positively span R^3.

To avoid considering all the possible combinations of 4 points on the object, an oriented search is performed looking for sets of points fulfilling condition P1, and condition P2 is tested on these sets of points to find the FC grasps. The algorithm has the following steps:

- The object's surface is discretized in a regular grid of voxels (cubic pixels) C_j (Figure 3.5). For each voxel there is a set of points T_j whose normal intersects the voxel. This intersection is found with the Bresenham‘s algorithm, commonly employed for drawing lines in computer graphics.

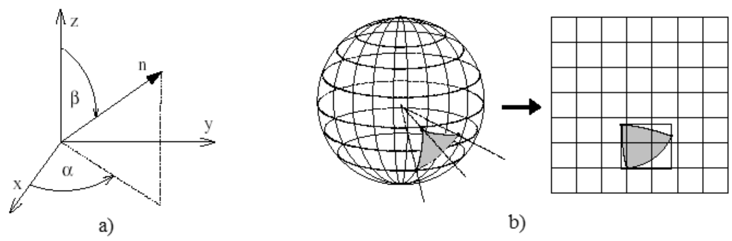

Figure 3.6: Concurrent grasps for discretized objects: a) Parametrization of the normal vectors, b) Mapping the unitary sphere to the cartesian plane (Source: [37]).

- The grasp search continues for the T_j's with at least four members, i.e. for voxels intersected by at least four friction axis cones (normal vectors), since there is a bigger chance to find an intersection point lying in the four friction cones. For each T_j considered, the inward normals of all the set are tested whether they positively span $\Re^3$; this condition is verified if the convex hull of the corresponding points contains the origin. This is a conservative approach, since there may be points whose normals do not positively span $\Re^3$ but whose friction cones allow a FC grasp; however, these grasps have a poor performance in front of external perturbations in some particular directions. The sets T_j failing this test are not considered for further processing.

- A predefined number of random points p are picked up inside the voxel C_j, and it is assumed that each point is the intersection point of four contact forces. With this assumption, the set $M \subset T_j$ of points whose friction cones contain the point p is generated (Figure 3.5); the test is performed by comparing the semi-angle of the friction cone with the angle between the cone's axis and the line between the point and the cone's vertex.

- All the possible combinations of 4 points in M fulfilling the condition P2 are listed, i.e. the possible combinations positively spanning $\Re^3$. For any three vectors, the fourth must lie strictly inside the thihedron formed by the inverses of the three given vectors (otherwise, they would lie in the same half space). It is important that every combination is listed without any repetition; to facilitate this ordering, each normal vector is parametrized using an ordered pair of angles (α, β), with $\alpha \in [0, 2\pi]$ the angle between the x-axis and the projection of the vector on the xy plane, and $\beta \in [0, \pi]$ the angle between the z axis and the vector (Figure 3.6). With this parametrization, a sorted order can be imposed by defining that a vector $\boldsymbol{v_a} = (\alpha_a, \beta_a)$ precedes $\boldsymbol{v_b} = (\alpha_b, \beta_b)$ when $\beta_a < \beta_b$, or when $\alpha_a < \alpha_b$ if $\beta_a = \beta_b$.

  An unitary vector can be represented as a point on the unitary sphere; a trihedron formed by three unitary vectors intersects the unitary sphere in a triangular spherical region. To simplify the search of point quadruples, the unitary sphere is mapped to a cartesian plane with the parameters (α, β), and the triangular spherical region is approximated by a rectangular region that encloses it (taking into account the singularities arising in $\alpha = 0$ and in the north and south poles). Thus, the algorithm searches for the points enclosed in the box; for each contained point, the candidate quadruple is tested for positive spanning of $\Re^3$.

- The algorithm is performed again with a finer discretization of C_j; the variation in resolution leads to find a wider set of FC grasps.

The implementation of the algorithm uses a random discretization of the object's surface, but the study of sampling techniques that may improve the algorithm's efficiency is not considered.

Table 3.1: Main references in grasp synthesis for arbitrary objects.

Reference	Object	No. fingers	Generated grasp
[24], 1993	Smooth surfaces	2 SC	Optimal grasp according to an energy function
[26], 1997	Bounding box of the object	3 PCWF	Best grasp among a group of heuristically generated grasps
[27], 1999	Arbitrary	4 PCWF Concurrent grasp	Best grasp among a group of heuristically generated grasps
[28], 2003	Piecewise smooth	n PCWF	Optimal grasp minimizing the Q distance, d_Q
[31], 2004	Piecewise smooth	n PCWF	Optimal grasp minimizing the function $f(\boldsymbol{u})$
[34], 2003	Arbitrary	3,4 and 5 PCWF	Best grasp among a group of heuristically generated grasps
[35], 2001	Discretized	7 FPC	One FC grasp
[36], 2004	Discretized	n FPC or PCWF	One FC grasp
[37], 2004	Discretized	4 PCWF Concurrent grasp	Set of grasps

Chapter 4

Empirical grasp synthesis and fixture design

4.1 Empirical approaches for grasp synthesis

In general, algorithms for analytic grasp synthesis are used in the planning of precision grasps, i.e. grasps where the fingertips are in contact and apply forces on the object. This kind of grasps is very useful in manipulation of objects; however, power grasps provide a better restriction on the object; in these grasps the palm and the proximal part of the fingers are also in contact with the object. The synthesis of power grasps relies mainly on empirical approaches, based on the study of the geometry that the human hand adopts to perform different tasks. The identification of the basic hand shapes in grasping is generically known as the taxonomy of grasping. Initially, grasps were classified in two large groups, power (volar) grasps and precision (non volar) grasps [38]. A more detailed classification uses six groups: cylindrical, spherical, lateral, pinch, hook and palm grasps [39]. Grasps used in manufacturing tasks are classified in [40]; an expert system is also developed to choose grasps based on the object shape and the task requirements. The taxonomies of grasping are summarized in [41], and a general classification is presented, based on the idea of virtual fingers (surfaces of the hand in contact with the object). Empirical grasp synthesis uses a variety of tools, such as fuzzy logic [42], neural networks [43] or knowledge-based systems [44, 45]. Some works in empirical grasp synthesis are summarized below.

A grasp planner for 3D objects with a Barrett hand is presented in [46]. The planner tries to imitate the human behavior: before grasping an object, humans unconsciously simplify the task to select one posture among a few options, depending on the object geometry and task to be performed. The grasp planning process has two stages. First, it generates a set of grasp starting positions, i.e. position and orientation of the hand and a grasp pre-shape selected for either a cylindrical or spherical grasp. The starting position is computed for a simplified model of the object's geometry, built from primitive shapes such as boxes, spheres and cones; each primitive shape is associated with different grasp strategies that generate candidate locations with greater probability to provide a high quality grasp. In the second stage, the grasping simulator GraspIt! [47] closes the fingers on the real model of the object, starting from the initial hand location, and subsequently evaluates the grasp quality. After all the generated possibilities have been tested, the system presents the best generated grasps. A similar two-stages procedure is used in [48]; first, it generates a preliminary grasp, optimum according to the hand restrictions; later, this preliminary grasp is used as the initial value in a search procedure of an optimal quality grasp.

Another approach uses a database with different FC grasp positions for a predefined object [49]. The database is generated off-line using a geometric simulation of the grasping hand and the object; different postures are generated by changing the object position and the angle of approach to the object, and the convex hull of the primitive contact wrenches and corresponding grasp quality are stored for each FC grasp tested. At runtime, the estimated mass distribution of the object (that may have variations with respect to the geometric model used to generate the database) and the information regarding force/moment that the object must resist, yield a single perturbation wrench, which is a point in the 6D wrench space. The appropriate grasp is selected from the database, looking for the grasp whose convex hull has a facet nearest the disturbance wrench, as this grasp will require minimum contact forces to resist the perturbation.

One successful grasp example can be the basis to generate a family of grasps with a quality at least as good as the quality from the example; such approach is presented in [50]. The example provides the approximated location of the independent contact regions; subsequently, the independent contact regions for each finger are computed. The algorithm is polynomial in the number of contacts, which makes it suitable for a large number of contacts; however, for objects with a small number of contacts there are better algorithms. The method is implemented for 5 to 12 frictional contacts (although it is also useful for frictionless contacts) and in a variety of tasks. The algorithm computes FC grasps, grasps with partial FC or simply grasps with a higher quality than the example's quality (this is specially suitable for non FC grasps computed to resist specific perturbations, for instance, the object's weight). Maximal independent contact regions and grasps with minimum contact forces are not generated, and the outcomes for the algorithm greatly depend upon the initial example. A previous version of this approach is found in [51]. A similar method which uses images from human grasps to extract the approximated location of the contact points as an example for a robotic hand is presented in [52].

4.2 Fixture design

Fixtures play an important role in manufacturing tasks such as assembly, inspection and machining of workpieces. The problem of fixture design is related to the grasp planning problem, as both of them generate contact locations to immobilize the object, i.e. a number of location elements (fixels or fingers) must be placed on the object to immobilize it (to fixture or grasp it). The fixture elements include passive elements (plates, V-blocks, locators) and active elements (clamps). A complete kinematic restriction (form-closure, total fixturing) is always desired in fixturing, with additional objectives such as the minimization of workpiece positional errors; force-closure is usually desired in grasp planning, as the object must resist external perturbation wrenches. Fixture design usually considers frictionless contacts, since the forces involved in manufacturing tasks are dynamical and have high magnitudes; on the other hand, frictional effects are usually considered in grasp synthesis.

The initial approach for 3D fixture design in [53] extends the fixture design algorithms for polygonal objects [54, 55] to prismatic objects with polygonal base (polyhedral prismatic workpieces), and proposes a modular fixture system with four contact points. A complete algorithm to enumerate the possible locations for the 4 contact points providing force-closure is also presented. The results for polyhedral prismatic workpieces are further generalized in [56] for 3D fixture design taking into account the task constraints. Later works are focused on polyhedral objects; for instance, in [57, 58] a modular fixture device is presented, and an algorithm to generate concurrent fixtures immobilizing the object with 4 frictional contacts is proposed.

Seven modular struts are used in [59] to compute form-closure fixtures on 3D objects. The algorithm enumerates all the 7 points subsets which provide form-closure, and sorts them according to a quality criterion, for instance the minimization of reaction forces under an external force specified by the user. Several basic criteria based on accessibility analysis to test the eligibility of a given surface that may be considered as a fixture surface are presented in [60]. A two-stages algorithm to generate a fixture configuration is presented in [61]. The first stage searches for a set of surfaces which may provide a form-closure fixture using the algorithm presented in [20] and already described in Subsection 2.4.3. The second stage determinates an optimal fixture with the solution of a quadratic programming problem; the objective function maximizes the euclidean distance between the fixture points and the center of mass of the object, and the linear restrictions are the form-closure conditions.

Several algorithms for fixture design in 3D generic objects have been presented. The synthesis of fixtures with 6 contact points and a clamp in a discrete domain (a cloud of points describing the surface of the object) is presented in [62], [63] and [64]; the algorithm minimizes the workpiece positional errors, an objective called D-optimality, by maximizing the determinant of a matrix M containing information on the contacts locations. Other optimality criteria include the A-optimality (maximization of the condition number of M) or the E-optimality (maximization of the minimum eigenvalue of M). The proposed algorithm iteratively changes the contact points, replacing in each iteration the contact giving the minor contribution to the desired objective, until it finds a fixture that maximizes the optimality criterion. The location and orientation for the clamp is predetermined in [62] and [63], but the clamp is selected after locating the 6 passive contacts in [64]. The algorithm finally returns a good but not optimal fixture, since an exhaustive search is not performed because of the large collection of points in the domain.

Fixtures provided by the previous algorithm are studied in [65], evaluating their performance with three criteria: D-optimality, norm of the locator contact forces and dispersion in the contact forces (in a minimum dispersion fixture all the contact forces in the fixture elements have similar magnitudes). The fixture performance characteristics are also studied, concluding that the most important criteria are the D-optimality and the dispersion of the locator contact forces.

A synthesis procedure similar to the algorithm presented in Subsection 3.2.1 is used in [66] for 3D fixture synthesis; the D-optimality is used to produce more robust grasps in front of workpiece positional errors.

Chapter 5

Conclusions

Grasp synthesis for 3D objects has been tackled with two different approaches: empirical or analytical. The empirical approach (knowledge-based) imitates human grasp using heuristics to choose a grasp shape from a set of basic hand postures. On the other hand, analytic approaches choose the finger positions and the hand configuration with kinematical and dynamical formulations, in general optimizing an objective function such as the grasp stability or the resistance to external perturbations. This technical report has reviewed the main analytical algorithms for grasp synthesis in polyhedral or arbitrarily shaped 3D objects. The relation between the fixture design for manufacturing tasks and the grasp synthesis has also been discussed.

The grasp synthesis for 3D complex objects, described with piecewise smooth surfaces or with a cloud of discrete surface points, is an active area of research. In this field, the following points can be tackled in future research:

- There are few algorithms to synthesize grasps in 3D discretized objects; new algorithms can be explored.
- The computation of independent contact regions for 3D arbitrary objects has not been tackled.
- Very few algorithms include the kinematical restrictions arising from the robotic hand in the grasp synthesis process.

Bibliography

[1] A. Bicchi. On the closure properties of robotic grasping. *Int. J. Robotics Research*, 14(4):319–344, 1995.

[2] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4):541–558, 1987.

[3] X. Markenscoff, L. Ni, and C.H. Papadimitriou. The geometry of grasping. *Int. J. Robotics Research*, 9(1):61–74, 1990.

[4] Y.H. Liu. Computing n-finger form-closure grasps on polygonal objects. *Int. J. Robotics Research*, 19(2):149–158, 2000.

[5] J. Cornellà and R. Suárez. Fast and flexible determination of force-closure independent regions to grasp polygonal objects. In *Proc. IEEE ICRA 2005*, pages 778–783, 2005.

[6] J. Cornellà and R. Suárez. On computing form-closure grasps/fixtures for non-polygonal objects. In *Proc. IEEE Int. Symp. Assembly and Task Planning, ISATP 2005*, pages 138–143, 2005.

[7] J. Cornellà and R. Suárez. Determining independent grasp regions on 2D discrete objects. In *Proc. IEEE/RSJ IROS 2005*, pages 2936–2941, 2005.

[8] R. Prado and R. Suarez. Prensiones con equilibrio de fuerzas en el espacio. Technical Report IOC-DT-P-2001-16, Technical University of Catalunya, 2001.

[9] V.D. Nguyen. Constructing force-closure grasps in 3D. In *Proc. IEEE ICRA 1987*, pages 240–245, 1987.

[10] V.D. Nguyen. Constructing force-closure grasps. *Int. J. Robotics Research*, 7(3):3–16, 1988.

[11] B. Mirtich and J. Canny. Easily computable optimum grasps in 2D and 3D. In *Proc. IEEE ICRA 1994*, pages 739–747, 1994.

[12] J. Ponce, S. Sullivan, J.D. Boissonat, and J.P. Merlet. On characterizing and computing three- and four-finger force-closure grasps for polyhedral objects. In *Proc. IEEE ICRA 1993*, pages 821–827, 1993.

[13] J. Ponce, S. Sullivan, A. Sudsang, J.D. Boissonat, and J.P. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *Int. J. Robotics Research*, 16(1):11–35, 1997.

[14] A. Sudsang and J. Ponce. New techniques for computing four-finger force-closure grasps of polyhedral objects. In *Proc. IEEE ICRA 1995*, pages 1355–1360, 1995.

[15] R. Prado and R. Suarez. Determinación de prensiones con equilibrio de fuerzas con cuatro puntos de contacto en objetos poliédricos. In *Proc. 1a Jornada de Recerca en Automàtica, Visió i Robòtica, AVR'04*, pages 277–283, 2004.

[16] Y.H. Liu, D. Ding, and S. Wang. Constructing 3D frictional form-closure grasps of polyhedral objects. In *Proc. IEEE ICRA 1999*, pages 1904–1909, 1999.

[17] D. Ding, Y.H. Liu, and S. Wang. Computation of 3-D form-closure grasps. *IEEE Trans. Robotics and Automation*, 17(4):515–522, 2001.

[18] D. Ding, Y. Liu, and S. Wang. Conmputing 3-D optimal form-closure grasps. In *Proc. IEEE ICRA 2000*, pages 3573–3578, 2000.

[19] D. Ding, Y. Liu, Y.T. Shen, and G.L. Xiang. An efficient algorithm for computing a 3D form-closure grasp. In *Proc. IEEE/RSJ IROS 2000*, pages 1223–1228, 2000.

[20] Y.H. Liu. Qualitative test and force optimization of 3-D frictional form-closure grasps using linear programming. *IEEE Trans. Robotics and Automation*, 15(1):163–173, 1999.

[21] D. Ding, Y.H. Liu, J. Zhang, and A. Knoll. Computation of fingertip positions for a form-closure grasp. In *Proc. IEEE ICRA 2001*, pages 2217–2222, 2001.

[22] T. Omata. Finger position computation for 3-dimensional equilibrium grasps. In *Proc. IEEE ICRA 1993*, pages 216–222, 1993.

[23] J. Hong, G. Lafferriere, B. Mishra, and X. Tan. Fine manipulation with multifinger hands. In *Proc. IEEE ICRA 1990*, pages 1568–1573, 1990.

[24] I.M. Chen and J.W. Burdick. Finding antipodal point grasps on irregularly shaped objects. *IEEE Trans. Robotics and Automation*, 9(4):507–512, 1993.

[25] A. Jaklic, A. Leonardis, and F. Solina. *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational Imaging and Vision*. Kluwer Academic Publishers, 2000.

[26] M. Fischer and G. Hirzinger. Fast planning of precision grasps for 3D objects. In *Proc. IEEE/RSJ IROS 1997*, pages 120–126, 1997.

[27] Ch. Borst, M. Fischer, and G. Hirzinger. A fast and robust grasp planner for arbitrary 3D objects. In *Proc. IEEE ICRA 1999*, pages 1890–1896, 1999.

[28] X. Zhu and J. Wang. Synthesis of force-closure grasps on 3-D objects based on the Q distance. *IEEE Trans. Robotics and Automation*, 19(4):669–679, 2003.

[29] D.G. Kirkpatrick, B. Mishra, and C.K. Yap. Quantitative Steinitz's theorem with applications to multifingered grasping. *Discrete and Computational Geometry*, 7(3):295–318, 1992.

[30] C. Ferrari and J. Canny. Planning optimal grasps. In *Proc. IEEE ICRA 1992*, pages 2290–2295, 1992.

[31] X. Zhu and H. Ding. Planning force-closure grasps on 3-D objects. In *Proc. IEEE ICRA 2004*, pages 1258–1263, 2004.

[32] X. Zhu and H. Ding. Computation of force-closure grasps: An iterative algorithm. *IEEE Trans. Robotics*, 22(1):172–179, 2006.

[33] X. Zhu, H. Ding, and M.Y. Wang. A numerical test for the closure properties of 3-D grasps. *IEEE Trans. Robotics and Automation*, 20(3):543–549, 2004.

[34] Ch. Borst, M. Fischer, and G. Hirzinger. Grasping the dice by dicing the grasp. In *Proc. IEEE/RSJ IROS 2003*, pages 3692–3697, 2003.

[35] D. Ding, Y.H. Liu, and M.Y. Wang. On computing inmobilizing grasps of 3-D curved objects. In *Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, pages 11–16, 2001.

[36] Y.H. Liu, M.L. Lam, and D. Ding. A complete and efficient algorithm for searching 3-D form closure grasps in the discrete domain. *IEEE Trans. Robotics*, 20(5):805–816, 2004.

[37] N. Niparnan and A. Sudsang. Fast computation of 4-fingered force-closure grasps from surface points. In *Proc. IEEE IROS 2004*, pages 3692–3697, 2004.

[38] J. Napier. The prehensile movements of the human hand. *J. Bone Joint Surg.*, 38B(4):902–913, 1956.

[39] C.L. Taylor and R.J. Schwartz. The anatomy and mechanics of the human hand. *Artificial Limbs*, 2:22–35, 1955.

[40] M.R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Trans. Robotics and Automation*, 5(3):269–279, 1989.

[41] T. Iberall. Human prehension and dexterous robot hands. *Int. J. Robotics Research*, 16(3):285–299, 1997.

[42] D. Bowers and R. Lumia. Manipulation of unmodeled objects using intelligent grasping schemes. *IEEE Trans. on Fuzzy Systems*, 11(3):320–329, 2003.

[43] F. Leoni, M. Guerrini, C. Laschi, D. Taddeucci, P. Dario, and A. Starita. Implementing robotic grasping tasks using a biological approach. In *Proc. IEEE ICRA 1998*, pages 2274–2280, 1998.

[44] G.A. Bekey, H. Liu, R. Tomovic, and W.J. Karplus. Knowledge-based control of grasping in robot hands using heuristics from human motor skills. *IEEE Trans. Robotics and Automation*, 9(6):709–722, 1993.

[45] T. Iberall, J. Jackson, L. Labbe, and R. Zampano. Knowledge-based prehension: Capturing human dexterity. In *Proc. IEEE ICRA 1988*, pages 82–87, 1988.

[46] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. Automatic grasp planning using shape primitives. In *Proc. IEEE ICRA 2003*, pages 1824–1829, 2003.

[47] A.T. Miller and P.K. Allen. GraspIt! a versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4):110–122, 2004.

[48] R.D. Hester, M. Cetin, C. Kapoor, and D. Tesar. A criteria-based approach to grasp synthesis. In *Proc. IEEE ICRA 1999*, pages 1255–1260, 1999.

[49] Y. Liu, G. Starr, J. Wood, and R. Lumia. Spatial grasp synthesis for complex objects using model-based simulation. *Industrial Robot*, 32(1):24–31, 2005.

[50] N.S. Pollard. Closure and quality equivalence for efficient synthesis of grasps from examples. *Int. J. Robotics Research*, 23(6):595–614, 2004.

[51] N.S. Pollard. Synthesizing grasps from generalized prototypes. In *Proc. IEEE ICRA 1996*, pages 2124–2130, 1996.

[52] S.B. Kang and K. Ikeuchi. Toward automatic robot instruction from perception - mapping human grasps to manipulator grasps. *IEEE Trans. Robotics and Automation*, 13(1):81–95, 1997.

[53] A.S. Wallack and J.F. Canny. Modular fixture design for generalized polyhedra. In *Proc. IEEE ICRA 1996*, pages 830–837, 1996.

[54] R.C. Brost and K.Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. In *Proc. IEEE ICRA 1994*, pages 535–542, 1994.

[55] R.C. Brost and K.Y. Goldberg. A complete algorithm for designing planar fixtures using modular components. *IEEE Trans. Robotics and Automation*, 12(1):31–46, 1996.

[56] R.C. Brost and R.R. Peters. Automatic design of 3-D fixtures and assembly pallets. In *Proc. IEEE ICRA 1996*, pages 495–502, 1996.

[57] J. Ponce. On planning immobilizing fixtures for three-dimensional polyhedral parts. In *Proc. IEEE ICRA 1996*, pages 509–514, 1996.

[58] A. Sudsang, J. Ponce, and N. Srinivasa. Algorithms for constructing inmobilizing fixtures and grasps of three-dimensional objects. In *Algorithmic Foundations of Robotics II*, pages 360–380. A.K. Peters, 1997.

[59] R. Wagner, Y. Zhuang, and K. Goldberg. Fixturing faceted parts with seven modular struts. In *Proc. IEEE ISATP 1995*, pages 133–139, 1995.

[60] J. Li, W. Ma, and Y. Rong. Fixturing surface accesibility analysis for automated fixture design. *Int. J. Prod. Res.*, 37(13):2997–3016, 1999.

[61] D. Ding, Y.H. Liu, M.Y. Wang, and S. Wang. Automatic selection of fixturing surfaces and fixturing points for polyhedral workpieces. *IEEE Trans. Robotics and Automation*, 17(6):833–841, 2001.

[62] M.Y. Wang. Automated fixture layout design for 3D workpieces. In *Proc. IEEE ICRA 1999*, pages 1577–1582, 1999.

[63] Y. Wang and D. Pelinescu. Precision localization and robust force closure in fixture layout design for 3D workpieces. In *Proc. IEEE ICRA 2000*, pages 3585–3590, 2000.

[64] M.Y. Wang. An optimum design for 3-D fixture synthesis in a point set domain. *IEEE Trans. Robotics and Automation*, 16(6):839–846, 2000.

[65] M.Y. Wang and D.M. Pelinescu. Optimizing fixture layout in a point-set domain. *IEEE Trans. Robotics and Automation*, 17(3):312–323, 2001.

[66] D. Ding, G. Xiang, Y.H. Liu, and M.Y. Wang. Fixture layout design for curved workpieces. In *Proc. IEEE ICRA 2002*, pages 2906–2911, 2002.