2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Extended Independent Contact Regions for Grasping Applications

Bao-Anh Dang-Vu, Maximo A. Roa, and Christoph Borst

***Abstract*— Independent contact regions have been proposed as a way to overcome possible errors in finger positioning for grasping an object. Efficient implementations for their computation have been developed, that even allow their usage in real-time telemanipulation applications. However, the main problems in the computation of contact regions are that they strongly depend on the initial grasp used as an starting point, and that for a given initial grasp there is not a unique set of contact regions. This paper analyzes the optimality of current approaches for ICR computation in 2D, where the optimal regions are still easily computable, and proposes an algorithm to obtain contact regions closer to the optimal ones. The approach is implemented and analyzed for 2D and 3D objects with any number of contact points.**

***Index Terms*— Grasp planning, contact regions**

I. Introduction

Grasp planning has been an active area of research in the last decades. Several algorithms have been proposed for computing contact points on the object surface that guarantee some desired property in the grasp, mainly force closure (FC), i.e. that the forces applied by the fingers ensure the object immobility [1]. Besides the contact points, grasp synthesis algorithms usually provide also a proper hand configuration (finger positions and hand pose relative to the object) that allows the execution of the desired grasp. A recent review summarizing the main works in this area is presented in [2]. Most of the approaches for planning grasps provide precise contact locations for the fingers on the object surface, but real mechanical hands can hardly assure that the exact contact points are reached due to different sources of uncertainty (e.g. a poorly estimated friction coefficient, or uncertainties in the current object location).

The computation of independent contact regions (ICRs) on the object boundary was introduced to provide robustness to finger positioning errors, such that if each finger i is positioned on its corresponding $\mathcal{ICR}_i$ an FC grasp is always obtained, independently of the exact location of each finger [3]. The determination of ICRs was initially solved for 4 frictionless contacts on 2D polygonal objects, and for 2 frictional contacts on polygonal and polyhedral objects [3]. The concept was later extended to 3-finger grasps of polygonal objects [4], and 4-finger grasps of polyhedral objects [5] by analyzing the configuration of the *grasp space*, i.e. the space representing all the possible grasps on the object surface. The geometrical construction of contact regions starting with an initial example was proposed as a way to create families of grasps that fulfill a desired property such as FC with a minimum desired grasp quality [6]. A practical implementation of such approach was later proposed for creating contact regions on discretized objects, using both frictionless and frictional contact models [7], [8].

All authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. {firstname.lastname}@dlr.de

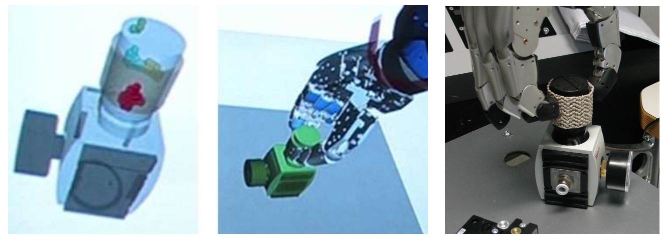


Fig. 1. Application of independent contact regions (ICRs) in a telemanipulation environment.

Most of the presented approaches for using or computing the ICRs are object centered, i.e. they do not include the hand kinematics in the computational loop. A procedure to obtain a hand configuration that reaches a predefined contact region on the object has been proposed [9], although it assumes that the specified contact regions are reachable for the mechanical hand. The related problem of how to compute reachable contact regions for a given object and hand kinematics has been efficiently solved, allowing the use of ICRs for applications in virtual reality and teleoperation, as shown in Fig. 1 [10], [11]. Real experiments involve different sources of uncertainty that influence the computation of the contact regions. A method for including the uncertainties in the location and normal orientation of the contact points that reduces the problem to choosing a suitable (and conservative) friction coefficient was recently proposed [12].

One of the main problems in the computation of ICRs is that the solution is not unique; for a given initial grasp, the analysis of the grasp space shows that different contact regions can be obtained. In this paper, we analyze optimality of current methods for computation of ICRs for 2D objects, based on a comparison with the full grasp space. Guided by this analysis, we propose a new method to extend the size of the ICRs, i.e. to create regions which are closer to the optimal regions for a given initial grasp. The method is inspired by a FC condition taken from classical grasp planning approaches, which allows an iterative extension of the ICRs. Although reachability of the computed regions is not explicitly considered in this paper, the proposed method can be easily merged with the reachability analysis previously presented in [10] to obtain extended regions that can also be realized by a given hand.

The required background and an efficient algorithm for computing independent contact regions is presented in Section II. Section III presents the approach for extending the contact regions initially computed, and provides the implementation details. Application examples are shown in Section IV. Finally, Section V discusses the contributions and future works along this line.

II. ICR Computation

A. Background

To find the contact regions, contacts between the fingers and the object are considered punctual. The force $\boldsymbol{f}_i$ applied at a point $\boldsymbol{p}_i$ generates a torque $\boldsymbol{\tau}_i = \boldsymbol{p}_i \times \boldsymbol{f}_i$ with respect to an object reference system located in the center of mass CM. $\boldsymbol{f}_i$ and $\boldsymbol{\tau}_i$ are grouped together in a wrench vector $\boldsymbol{\omega}_i = (\boldsymbol{f}_i \; \boldsymbol{\tau}_i)^T$. Friction between fingertips and object is described with Coulomb's law. In the 3-dimensional physical space this model is nonlinear and, to simplify the computations, the friction cone is linearized using an m-side polyhedral convex cone (the more sides, the better the approximation, but the higher the computational cost of dealing with the linearized cone). A wrench $\boldsymbol{\omega}_{ij}$ generated by a unitary force $\boldsymbol{f}_i$ along an edge of the linearized friction cone, i.e. $\boldsymbol{f}_i = \hat{\boldsymbol{n}}_{ij}$, is called a *primitive wrench*.

Each contact point $\boldsymbol{p_i}$ has m corresponding primitive wrenches $\boldsymbol{\omega}_{ij}$. A grasp is defined by the set of contact points $\mathcal{G} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$, and thus associated to the set $\mathcal{W} = \{\boldsymbol{\omega}_{11}, \ldots, \boldsymbol{\omega}_{1m}, \ldots, \boldsymbol{\omega}_{n1}, \ldots, \boldsymbol{\omega}_{nm}\}$. Therefore, $\mathcal{G}$ and $\mathcal{W}$ will be used as representative sets of a particular grasp.

A grasp is force closure (FC) if and only if the origin O of the wrench space lies strictly inside the convex hull of $\mathcal{W}$, represented as $CH(\mathcal{W})$ [1]. In this work, the condition $O \in CH(\mathcal{W})$ is checked by verifying that O and the centroid P of the primitive contact wrenches in $\mathcal{W}$ (which is always an interior point of $CH(\mathcal{W})$) lie on the same side of the hyperplane H_k containing the facet k of $CH(\mathcal{W})$, $\forall\, k$ [7].

The grasp quality is quantified with the largest perturbation wrench that the grasp can resist independently of the perturbation direction [13]. This grasp quality is equivalent to the radius of the largest hypersphere centered on O and fully contained in $CH(\mathcal{W})$, i.e. it is the distance from O to the closest facet of $CH(\mathcal{W})$.

B. Algorithm for ICR computation

Two main approaches have been proposed to compute in an efficient way the independent contact regions, both relying on geometrical constructions on the wrench space [7], [8]. The approach presented in [7] creates a conservative search space that leads to a quick computation of the regions around the initial grasp. The approach presented in [8] considers a more complete condition for computing the contact regions that includes the linear combination of the primitive wrenches for a given contact point, but is computationally very expensive due to an inclusion test that repetitively solves a linear programming problem. The method presented here is a new algorithm for ICRs computation, which also considers the linear combination of primitive wrenches but relies on simple geometrical conditions that allow a quick computation of the regions.

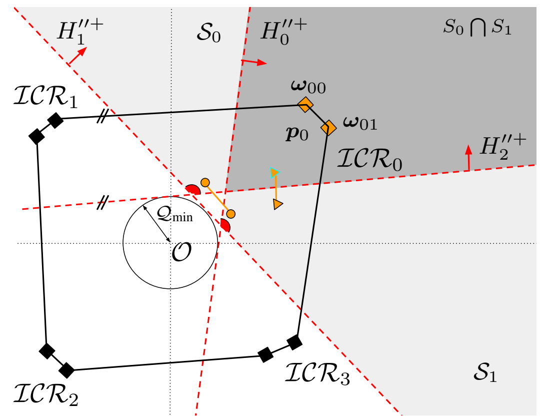


Fig. 2. ICR computation in an abstract 2-dimensional wrench space. For the point $\boldsymbol{p}_0$, the relevant supporting hyperplanes and corresponding search regions are shown. Two cases of valid primitive wrenches are considered: at least one primitive wrench in the area $S_0 \bigcap S_1$ (orange triangles) and at least one primitive wrench in each search region S_i (orange circles). All the points with valid primitive wrenches form the region $\mathcal{ICR}_0$.

The procedure for finding $\mathcal{ICR}_0$ for a 4-finger frictional grasp is illustrated in Fig. 2 with an abstract 2-dimensional wrench space that includes 2 primitive wrenches per contact point. Similar to previous methods, this one relies on a given starting FC grasp $\mathcal{G}$ with initial grasp quality Q_0. The approach builds the hyperplanes H_k'' parallel to the supporting hyperplanes H_k, and tangent to the task wrench space, represented by a sphere with radius αQ_0, $0 < \alpha \leq 1$. Let $\mathcal{K}_i = \{H_k \mid \exists\, \boldsymbol{\omega}_{ij} \subset H_k\}$, i.e. the set of supporting hyperplanes of the convex hull $CH(\mathcal{W})$, such that each H_k contains at least one primitive wrench of the contact point i. Each hyperplane in $\mathcal{K}_i$ creates a positive and negative half-space, defined such that the origin O of the grasp wrench space lies on the negative half-space. A search region S_k is defined as the positive half space $H_k''^+$. The method uses a tree-structure representing the local mesh around each contact point $\boldsymbol{p}_i$. A simple breadth-first-search is performed to build each $\mathcal{ICR}_i$ by looking for the neighbor points of $\boldsymbol{p}_i$ such that at least one of its primitive wrenches lies on each search region S_k. The points that fulfill this condition belong to the region $\mathcal{ICR}_i$.

The formal pseudo-code of the ICR algorithm is presented in Algorithms 1 and 2.

Complexity: The computation of the convex hull in the first step of the algorithm uses the Qhull-package [14]. For a six-dimensional input, it has a complexity of $\mathcal{O}((n_v \cdot m)^3/6)$ where n_v is the number of contact points whose associated m primitive wrenches form the vertices of $CH(\mathcal{W})$. As for the breadth-first-search, since for each contact point each neighbor is inspected, the complexity is $\mathcal{O}(N \cdot n)$, with N the total number of points describing the object. Hence, the overall complexity is $O(((n_v \cdot m)^3/6) + N \cdot n)$.

Algorithm 1: ICR computation

Given:
- an object discretized in points $\mathcal{O} = \{\boldsymbol{p}\}$
- a n-finger force closure grasp $\mathcal{G} = \{\boldsymbol{p}_i\}_{i=1,\cdots,n}$ with associated grasp quality Q_0
- a factor α ($0 < \alpha \leq 1$) defining the minimum quality αQ_0 for the ICRs

Output: $\mathcal{ICR} = \{\mathcal{ICR}_1, \ldots, \mathcal{ICR}_n\}$

Compute $CH(\mathcal{W})$
$\mathcal{K} = \{\mathcal{K}_i\}_{i=1,\cdots,n} \leftarrow$ All the supporting hyperplanes of $CH(\mathcal{W})$ that contain at least one primitive wrench for each contact point $\boldsymbol{p}_i$
$\mathcal{K}'' \leftarrow$ Set of hyperplanes parallel to $\mathcal{K}$ and tangent to the insphere of radius αQ_0
foreach *contact point* $\boldsymbol{p}_i$ **do**
  Create a queue $\mathcal{Q}$
  Enqueue $\boldsymbol{p}_i$ onto $\mathcal{Q}$
  Mark $\boldsymbol{p}_i$
  $\mathcal{K}''_i \leftarrow$ Set of parallel hyperplanes of $\mathcal{K}''$ for the contact point $\boldsymbol{p}_i$
  while $\mathcal{Q}$ *is not empty* **do**
    $\boldsymbol{p}_t \leftarrow \mathcal{Q}.dequeue()$
    if *inclusionTest* ($\boldsymbol{p}_t$, $\mathcal{K}''_i$) ; /* Algorithm 2 */
    then
      $\mathcal{ICR}_i \leftarrow \boldsymbol{p}_t$
      foreach $\boldsymbol{p}_v$ *neighbor of* $\boldsymbol{p}_t$ **do**
        if $\boldsymbol{p}_v$ *is not marked* **then**
          Mark $\boldsymbol{p}_v$
          Enqueue $\boldsymbol{p}_v$ onto $\mathcal{Q}$

Algorithm 2: inclusionTest for ICR

Given:
- the tested point $\boldsymbol{p}_t$
- the shifted hyperplanes $\mathcal{K}''_i$ that contain at least one primitive wrench of $\boldsymbol{p}_i$

Output: a boolean (true if $\{\forall H''_k \in \mathcal{K}''_i, \exists \boldsymbol{\omega}_{tj} \in H''^{+}_k, j = 1, \ldots, m\}$)

foreach $H''_k \in \mathcal{K}''_i$ **do**
  contain $\leftarrow$ false
  $H''^{+}_k \leftarrow$ positive side of H''_k
  foreach *primitive wrench* $\boldsymbol{\omega}_{tj}$ *of* $\boldsymbol{p}_t$ **do**
    if $\boldsymbol{\omega}_{tj} \subset H''^{+}_k$ **then**
      contain $\leftarrow$ true
      break ; /* go to next plane */
  if *contain is false* **then**
    return false
return true

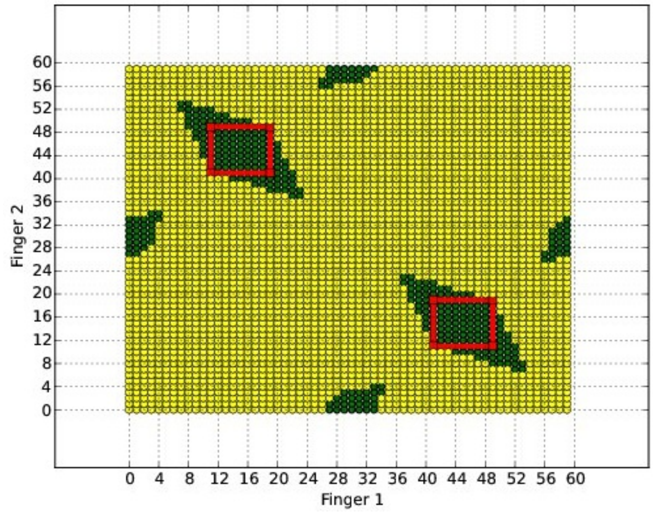


Fig. 3. 2-dimensional grasp space for an ellipse grasped with 2 frictional fingers. The FC space is shown in dark green, and the non-FC space in light yellow. The two largest rectangles depicted in red correspond to the largest ICRs that can be obtained on the object.

C. Exploration of the grasp space

The *grasp space* for a given object is considered as the p-dimensional space defined by the p parameters that represent the positions of possible contact points on the object. A point in the grasp space represents a possible grasp $\mathcal{G}$ on the object surface. Therefore, the grasp space is divided into two complementary subsets: the FC space, formed by the points that correspond to FC grasps, and the non-FC space, whose points correspond to non-FC grasps.

To study the performance of the ICR algorithm, we propose to find the *optimal* independent contact regions. For each finger, its ICR represent a continuous region on the surface of a discretized object. Since one axis of the grasp space represents the possible positions of a finger on the discretized object, one ICR can be shown as a line in the grasp space. As a consequence, all the ICRs, defining all the possible combinations of FC grasps, are represented as a rectangle in the case of two fingers, or a p-parallelepiped in the more general case of p fingers. The optimal ICRs are then defined as the largest p-parallelepiped in the grasp space that contains the initial grasp $\mathcal{G}$.

The proposed method to find the optimal ICRs is based on the problem of finding the largest rectangle in a 2-dimensional binary matrix with a dynamic programming algorithm. Let the grasp space be represented as a p-binary matrix containing only 0's (representing the non FC grasps) and 1's (for FC grasps). Finding the largest p-parallelepiped in the FC grasp space is equivalent to find the largest p-parallelepiped containing only 1's.

The main idea is to build a histogram that keeps track of the consecutive 1's, and then to hold a stack that dynamically keeps the information of potential rectangles, in decreasing order. The stack holds the height of each potential rectangle. By maintaining a stack in decreasing order of size, the maximum area can be easily calculated in one scan. The

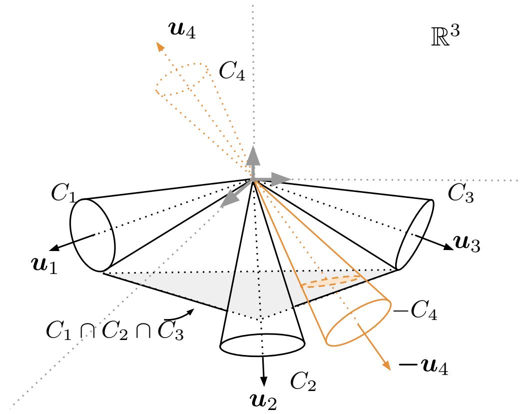


Fig. 4. Condition for a FC grasp by [5]: Four vectors θ-positively span $\mathbb{R}^3$. The intersection of the trihedra formed by all triples of vectors belonging to the cones C_1, C_2 and C_3 is shown in grey. The cone C_4 is depicted in orange and its opposite lies in the intersection.

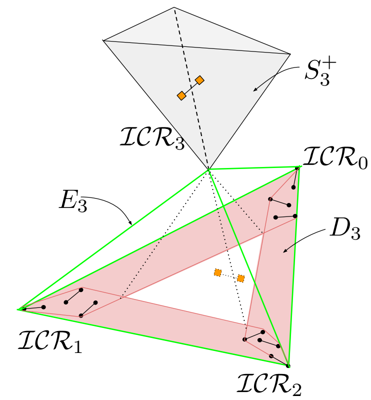


Fig. 5. Representation of the different regions for the extension of $\mathcal{ICR}_3$ from a 4-finger grasp. $\mathcal{ICR}_3$ is depicted in orange square, and its reflection is dashed. The external region E_3 is represented in green, the discard region D_3 in red, and the search region S_3^+ (the reflection of $E_3 - D_3$) in grey.

computation of the largest parallelepiped in p dimensions is done then by reducing the problem to $p-1$ dimensions.

Complexity: Let N the total number of elements and k be the number of elements in one dimension of the grasp space. A brute-force method runs in $\mathcal{O}(N^2)$ in 2D. The presented method runs in linear time $\mathcal{O}(N)$ in 2D and $\mathcal{O}(k^{2(p-1)})$ otherwise. Note that the complexity increases rapidly with the dimension of the space.

III. Extending ICRs

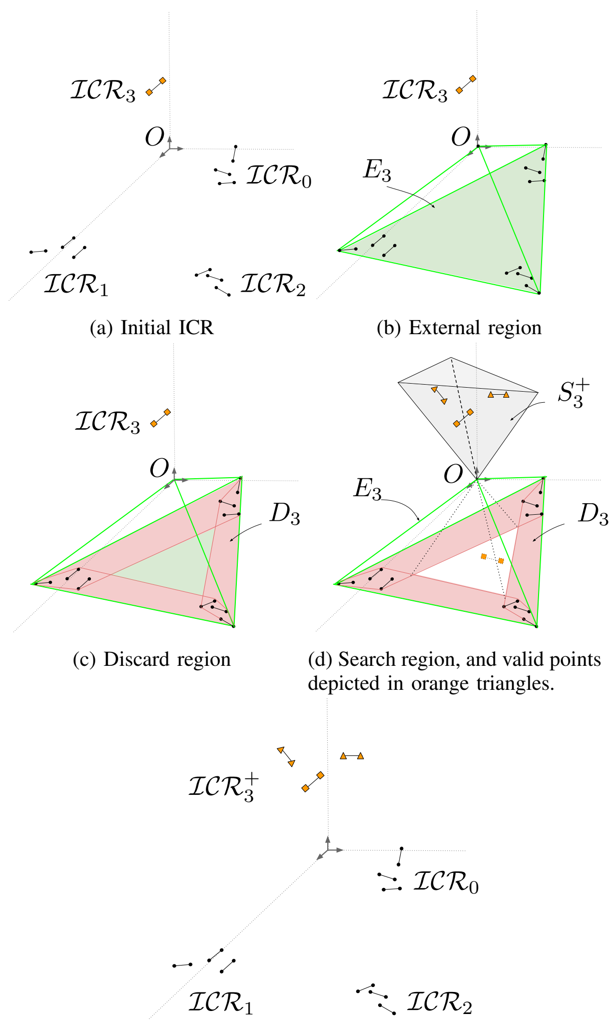
Starting with a precomputed set of ICRs, the method presented in this section extends as much as possible the initial regions. The approach is inspired by the necessary and sufficient condition proposed in [5] for finding an n-finger frictional FC grasp, as illustrated in Fig. 4:

Definition 3.1: n vectors (representing the generatrices of n cones) θ-positively span $\mathbb{R}^{n-1}$ when, for any $(n-1)$-tuple of these vectors, the n^{th} cone centered on the direction opposite of the n^{th} vector lies in the interior of the intersection of the convex polyhedron formed by all $(n-1)$-tuple of vectors belonging to their cones.

An application to the more complex case of n-finger ICRs is possible if the iterative extension of each ICR is considered, and the cones in the previous definition are replaced by convex cones for each ICR. To find the interior of the convex polyhedron mentioned in the previous definition, a discard region D_i is substracted from an external region E_i, as shown in Fig. 5. Thus, for extending $\mathcal{ICR}_i$, a search region S_i^+ is defined as the reflection of the previous substraction, i.e. $S_i^+ = \overline{(E_i - D_i)}$.

Fig. 6 represents the different steps of the extension algorithm for one ICR in a 4-finger frictional grasp, starting with a given set of ICRs (Fig. 6a), following with the construction of a search region (Fig. 6b to Fig. 6d), and the inclusion test that defines if a point is inside the new ICR.

1) Construction of a set of primitive wrenches for the search region: The computation of the initial ICRs with Algorithm 1 guarantees that each point included in an ICR



(a) Initial ICR

(b) External region

(c) Discard region

(d) Search region, and valid points depicted in orange triangles.

(e) Result of the extension as $\mathcal{ICR}_3^+$

Fig. 6. Steps in the extension of $\mathcal{ICR}_3$ in a conceptual example. The ICRs of a 4-finger frictional grasp are represented. Wrenches belonging to $\mathcal{ICR}_3$ are depicted as orange squares and triangles.

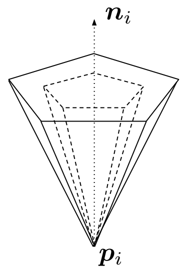


Fig. 7. Linearization of the friction cone around the normal $\boldsymbol{n}_i$ at a contact point $\boldsymbol{p}_i$. The friction cone represented is linearized in $n_e = 5$ edges and $n_l = 2$ layers.

has one of its primitive wrenches inside the intersection of all the search regions, or at least one primitive wrench in each search region.

Since the extension starts with the initial ICRs, the primitive wrenches from each point of the initial ICRs are needed to build the search region S_i^+. The last condition for the initial inclusion test doesn't assure that one primitive wrench is strictly inside the intersection of all the search regions, it implies that we could overestimate the external region E_i and the discard region D_i by using all the primitive wrenches from the initial ICR. Also, the external and discard regions are based one the construction of a convex hull: too few primitive wrenches might lead to situations where the convex hull does not exist.

Therefore, to avoid the overestimation of the external and discard regions for a given $\mathcal{ICR}_i$, we define the set $\mathcal{M}_i$ of primitive wrenches that lie strictly inside the intersection of all the search regions S_i for each plane H_k'':

$$\mathcal{M}_i = \left\{ \boldsymbol{\omega}_{rj} \mid \forall \boldsymbol{p}_r \in \mathcal{ICR}_i, w_{rj} \subset \bigcap_{k=1,\cdots,l} H_k''^{+} \right\} \quad (1)$$

To guarantee enough primitive wrenches to build the required convex hulls, we approximate the complete volume of the friction cone as n_l layers of a polyhedral convex cone of n_e edges, as represented in Fig. 7. Then, we have $n_\omega = n_e \times n_l$ primitive wrenches per contact point in each ICR.

Algorithm 3 details the construction of the set $\mathcal{M}_i$ of primitive wrenches for a given $\mathcal{ICR}_i$.

2) Search region: Considering a set of points $\mathcal{P}$, let $CC(\mathcal{P})$ be the convex cone of $\mathcal{P}$ that has a vertex in the origin O, and let $\mathcal{I}_i' = \{i' \in [1, \ldots, n] \mid i' \neq i\}$ be the set of all the indices except the i^{th}.

To obtain the search region S_i^+, we need to build the external region E_i and the discard region D_i. The external region E_i (Fig. 6b) is defined as the convex cone built from the set $\mathcal{M}_i' = \left\{ \bigcup_{i' \in \mathcal{I}'} \mathcal{M}_{i'} \right\}$ that is formed by the primitive wrenches of the ICRs except the i^{th} (which is going to be extended), i.e. $E_i = CC(\mathcal{M}_i')$

The discard region D_i (Fig. 6c) is defined based on the convex cone E_i. Since a facet of E_i contain primitive wrenches coming from different ICRs, we define the operator $\mathbf{comb}(f)$ such that for each facet f of E_i, it returns the union of the sets $\mathcal{M}_i$ for all the ICRs that contribute with at least one primitive wrench to f (Fig. 8).

Algorithm 3: UpdateMap

Given:
- the region $\mathcal{ICR}_i$ of the point $\boldsymbol{p}_i$, $\mathcal{ICR}_i = \{p_r\}$
- $\mathcal{K}_i'' = \{H_k''\}_{k=1,\cdots,l}$ the set of shifted hyperplanes for $\boldsymbol{p}_i$

Output: Set of primitive wrenches $\mathcal{M}_i$ for the region $\mathcal{ICR}_i$, such that all the primitive wrenches are inside $\bigcap_{k=1,\cdots,l} H_k''^{+}$

foreach $\boldsymbol{p}_r \in \mathcal{ICR}_i$ **do**
    foreach *primitive wrench* $\boldsymbol{\omega}_{rj}$ *of* $\boldsymbol{p}_r$ **do**
        if $\boldsymbol{\omega}_{rj} \subset \bigcap \mathcal{H}_k''^{+}$ **then**
            $\mathcal{M} \leftarrow \boldsymbol{\omega}_{rj}$
return $\mathcal{M}$

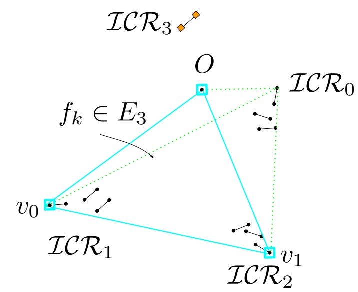


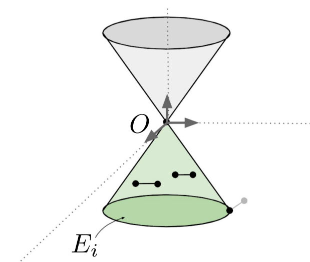
Fig. 8. Combination of ICRs for the facet f_k. $f_k = \{O, v_0, v_1\}$. The vertice v_0 belongs to $\mathcal{ICR}_1$ and v_1 to $\mathcal{ICR}_2$. As a result, $comb(f) = \{\mathcal{M}_1, \mathcal{M}_2\}$.

$$\mathbf{comb}(f) \quad = \quad \{\mathcal{M}_{i' \in \mathcal{I}'} \mid \exists \mathcal{M}_{i'} \supset \boldsymbol{\omega}_{i'j}, \forall \boldsymbol{\omega}_{i'j} \in f\} \quad (2)$$

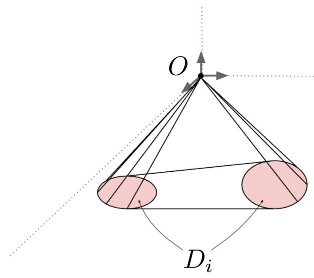
The discard region is then the union of the convex cones formed by all the possible combinations of ICRs defined by the facets of E_i:

$$D_i \quad = \bigcup_{\forall f \in E_i} CC\left(\mathbf{comb}\left(f\right)\right) \quad (3)$$

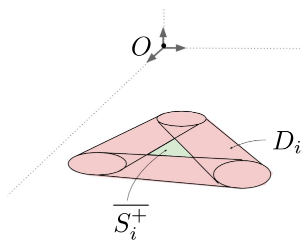
However, the discard region can take very different shapes depending on the number of ICRs considered for building the region, as represented in Fig. 9. The region D_i does not exist when the set $\mathcal{I}'$ contains only one index, which happens for a 2-finger grasp. It can also be made of disjoint regions, in the case that there is only two indices $\mathcal{I}'$, i.e. a 3-finger grasp, or it can be properly defined when the number of indices in $\mathcal{I}'$ is greater or equal to 3, i.e. for a grasp with minimum 4 fingers. Besides, D_i also does not exist when the origin is inside $CH(\mathcal{M}_i')$. In consequence, the search region S_i^+ is defined according to the different cases. Let $\overline{E_i}$ be the



(a) 2 fingers: $D_i = \emptyset$, $\overline{S_i^+} = E_i$ is depicted in green



(b) 3 fingers: D_i, represented in red, is disjoint



(c) 4 fingers: The search region in green is fully contained into E_i and surrounded by D_i in red.

Fig. 9. Existence of discard regions D_i depending on the number of contact points.

reflection of an E_i, and $\overline{CH(\mathcal{M}'_i)}$ the reflection of $CH(\mathcal{M}'_i)$,

$$S_i^+ = \begin{cases} \overline{E_i}, & n = 2 \ (\nexists D_i) \\ \overline{CH(\mathcal{M}'_i)}, & O \in CH(\mathcal{M}'_i) \ (\nexists D_i) \\ \overline{E_i - D_i}, & \exists D_i \end{cases} \tag{4}$$

3) Inclusion test: As presented in Eq. 4, the search region depend on the existence of the discard region, which in turn depends on the number of contact points in the initial grasp (Fig. 9). Any added point to the extended ICR must guarantee that an FC grasp is possible when choosing any point from all the other ICRs to create a grasp. Depending on the different cases, a point helps to build a convex hull that contains the origin when the following geometric conditions are fulfilled:

- For the case where no discard region exists, if only the linear combination of its primitive wrenches is inside S_i^+, while none of its primitive wrenches are inside S_i^+.
- For a 3-finger grasp the discard region D_i exists but is discontinuous (Fig. 9b), then a point is valid if at least one primitive wrench is inside S_i^+ and none is inside $\overline{D_i}$.
- For more fingers, S_i^+ is fully contained into $\overline{E_i}$ and surrounded by $\overline{D_i}$ (Fig. 9c), then a point is included into the region if all the primitive wrenches are inside S_i^+.

The complete algorithm for the extension of ICRs is provided in Algorithm 4, and the inclusion test is detailed in Algorithm 5.

Complexity: The complexity of the algorithm depends on the convex hull computation. It runs in $\mathcal{O}(r^3/6)$, where r is the number of points whose associated wrenches form the vertices of $CH(\mathcal{M}'_i)$. The number of call to the convex hull computation depends on the number of combinations defined

---

Algorithm 4: Extended ICR computation

Given:
- an object discretized in points $\mathcal{O} = \{\boldsymbol{p}\}$
- initial ICRs=$\{\mathcal{ICR}_i\}_{i=1,\cdots,n}$

Output: extended $\mathcal{ICR}^+$ ($\mathcal{ICR} \subset \mathcal{ICR}^+$)

Build the set of primitives wrenches $\mathcal{M}$ as defined in Algorithm 3.
Initialize $\mathcal{ICR}^+$ with $\mathcal{ICR}$
$\mathcal{I}_{\text{order}} \leftarrow$ Indices of $\mathcal{ICR}^+$ from the smallest to the largest size (number of points in the ICR).
foreach $\mathcal{ICR}_i^+, i \in \mathcal{I}_{order}$ **do**
  Create a queue $\mathcal{Q}$
  Enqueue the points of $\mathcal{ICR}$ onto $\mathcal{Q}$
  Mark the points
  Generate the set $\mathcal{M}'_i = \left\{ \bigcup_{i' \in \mathcal{I}'} \mathcal{M}_{i'} \right\}$
  Build the search region S_i^+ following Eq. 4
  while *$\mathcal{Q}$ is not empty* **do**
    $\boldsymbol{p}_t \leftarrow \mathcal{Q}.dequeue()$
    if *inclusionTest($\boldsymbol{p}_t, S_i^+$)* ; /* Algorithm 5 */
    then
      $\mathcal{ICR}_i^+ \leftarrow \boldsymbol{p}_t$
      foreach $\boldsymbol{p}_v$ *neighbour of* $\boldsymbol{p}_t$ **do**
        if $\boldsymbol{p}_v$ *is not marked* **then**
          Mark $\boldsymbol{p}_v$
          Enqueue $\boldsymbol{p}_v$ onto $\mathcal{Q}$

Algorithm 5: inclusionTest for ICR+

Given:
- the tested point $\boldsymbol{p}_t$, with primitive wrenches $\boldsymbol{\omega}_{tj}$, $j = 1, \ldots, m$
- the search region S_i^+ for the contact point $\boldsymbol{p}_i$

Output: a boolean

if $\nexists \boldsymbol{\omega}_{tj} \subset D$, **then**
  case $n = 2$
    if $\forall H_k \in CH(\mathcal{M}'_i), \exists \boldsymbol{\omega}_{tj} \in H_k^+$ *and* $\nexists \boldsymbol{\omega}_{tj} \subset E$ **then**
      return true
  case $n = 3$
    if $\exists \boldsymbol{\omega}_{tj} \subset S_i^+$ **then**
      return true
  case $n \geq 4$
    if $\forall \boldsymbol{\omega}_{tj}, \boldsymbol{\omega}_{tj} \subset S_i^+$ **then**
      return true

in Eq. 2. The theoretical maximal number of combinations for a set of $n-1$ elements is $\sum_{j=0,\cdots,n-1} \binom{n}{j} = 2^{n-1}$. Thus, for n fingers, the convex hull computation is called $n \times 2^{n-1}$ times.

IV. Numerically evaluated results

The proposed algorithms were implemented in C++ and tested on a Desktop Linux PC. For showing the performance of the algorithm, both 2D and 3D objects were used.

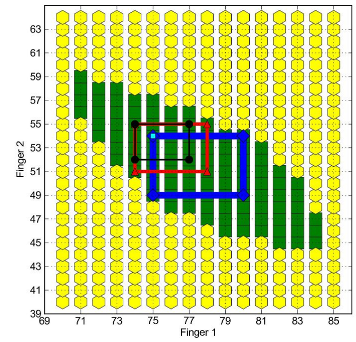
2D objects: To demonstrate the behavior of the extension algorithm, different computations of ICRs are performed for two different 2D objects, an ellipse discretized in 60 points, and a curve of polar equation $3/(1+3\cdot\cos(0.5\theta))$ discretized in 129 points. Table I summarizes the results for 400 random trials with 2, 3 and 4 fingered frictional grasps. The friction cone was linearized in 6 edges with a frictional coefficient of 0.45. In the case of 2 and 3 fingers, the computation of the ground-truth optimal ICRs (Section II-C) was still doable in a reasonable time, therefore the size of the ICRs is compared (in percentage) to the optimal size of ICRs for that initial grasp. The size of the ICRs is measured as the volume of the parallelepiped in the grasp space; the larger the volume, the more grasp possibilities that lead to a FC grasp on the object. Note that in general the obtained ICR extension depends on the number of fingers; the more fingers, the less extension it is possible to get. However, computational times do not increase beyond 8X for the maximum extension obtained.

Two examples of this benchmark are represented for a 2 finger grasp in Fig. 10 for the polar curve, and for a 3 finger grasp on the ellipse in Fig. 11.

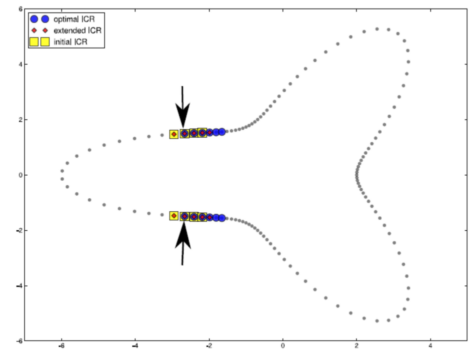
3D objects: Table II shows the results of ICR computations for two 3D objects, a parallelepiped discretized with 3422 points and a bottle discretized with 6834 points, for 400 random trials with 3 and 4 fingered frictional grasps. For 3D, the friction cone was linearized in 6 edges and 3 layers for both objects. The frictional coefficient used for the parallelepiped is 0.2 and for the bottle 0.6. Note that in the 3D case the time required to get the extended ICRs might be very high for online grasp computations, although the total size of the ICRs is increased in general above 2X.

V. Conclusions and discussion

In this work, two main problems in the computation of contact regions have been pointed out: the dependence on the initial FC grasp used as a starting point, and the non-uniqueness of the obtained solutions. To study the optimality of the computation of independent contact regions, a new algorithm for their computation is proposed, followed by an algorithm that computes extended ICRs which are closer to the optimal contact region as analyzed in the grasp space. The main drawback of the proposed method is that there is still a strong dependence of the initial FC grasp used as a starting point for computing the ICRs, which limits the ability to explore the complete wrench space. Therefore, although the new method increases the size of the contact regions, it still does not lead to the optimal size in the grasp

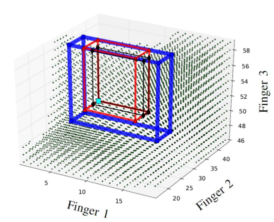


(a) Local grasp space around the initial grasp shown as a cyan point. In black the initial ICR, in red the extended ICR, and in blue the optimal ICR.

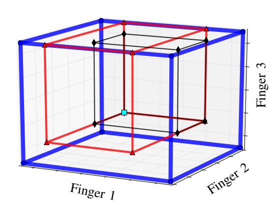


(b) ICRs on the polar curve. The initial ICR is depicted in yellow squares, the extended ICR in blue circles and the optimal ICR is shown in red diamonds.

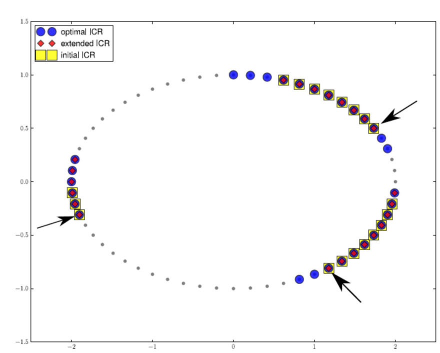
Fig. 10. Comparison of ICRs computation for a two-finger grasp on a polar curve. The initial, extended and optimal ICRs are shown in the local grasp space around the initial grasp (a) and on the object (b).



(a) Local force-closure grasp space around the initial grasp, depicted as a cyan point.



(b) Zoom: In black the initial ICR, in red the extended ICR, and in blue the optimal ICR.



(c) Representation on the ellipse

Fig. 11. Comparison of ICRs computation for a 3 finger grasp on an ellipse. The initial, extended and optimal ICRs computation are represented in the local grasp space around the initial grasp (a), (b) and on the object (c).

TABLE I
RESULTS FOR ICR EXTENSION ON 2D OBJECTS

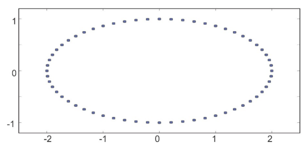
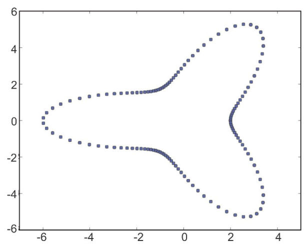
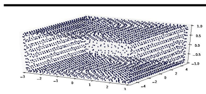
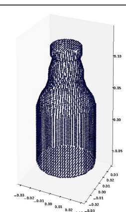
	No.fingers		Initial ICR	Extended ICR	Optimal ICR
	2	volume	13.2 (38%)	31.25 (91%)	34.1725 (100%)
		time (s)	$5 \cdot 10^{-4}$ ($\times 1$)	1 $\cdot 10^{-3}$ ($\times 2$)	4.6 10^{-2}
	3	volume	329.71 (27%)	662.71 (54%)	1229.83 (100%)
		time (s)	6 $\cdot 10^{-4}$ ($\times 1$)	1 $\cdot 10^{-3}$ ($\times 1.6$)	1.51 $\cdot 10^{2}$
	4	volume	7979.91 ($\times 1$)	8486.77 ($\times 1.06$)	
		time (s)	5 $\cdot 10^{-4}$ ($\times 1$)	5 $\cdot 10^{-4}$ ($\times 1$)	
	2	volume	14.65 (58%)	23.94 (94%)	25.47 (100%)
		time (s)	$1.75 \cdot 10^{-4}$ ($\times 1$)	$1.47 \cdot 10^{-3}$ ($\times 8$)	$2.06 \cdot 10^{-1}$
	3	volume	830.64 (40%)	973.215 (44%)	2432.16 (100%)
		time (s)	$8.0 \cdot 10^{-4}$ ($\times 1$)	$1.35 \cdot 10^{-3}$ ($\times 1.7$)	$4.67 \cdot 10^{2}$
	4	volume	32502.7 ($\times 1$)	37993.4 ($\times 1.16$)	
		time (s)	$8.0 \cdot 10^{-4}$ ($\times 1$)	$1.86 \cdot 10^{-3}$ ($\times 2$)	

TABLE II
RESULTS FOR ICR EXTENSION ON 3D OBJECTS

	No. fingers		Initial ICR	Extended ICR
	3	volume	26.8	543.28 ($\times 20$)
		time (s)	0.002	0.1804 ($\times 90$)
	4	volume	1.51e7	4.14e7 ($\times 2.7$)
		time (s)	0.40	4.31 ($\times 10$)
	3	volume	6069.37	12706.7 ($\times 2.1$)
		time (s)	0.0279	0.16 ($\times 6$)
	4	volume	7891.56	14204.8 ($\times 1.8$)
		time (s)	0.04	1.81 ($\times 45$)

space. As stated in the introduction, the reachability of the computed regions was not considered. However, the modular structure of the approach presented in [10] allows a seamless integration of the extended regions into the algorithm that computes reachable contact regions.

Although the method for extending the ICRs is expensive from the computational point of view in the case of 3D objects, it might still be useful for offline computations, where a database of grasps is preprocessed for cases where the complete 3D model of the object is known in advance.

The problem of dependency of the ICRs on the initial starting FC grasp is still unsolved. One potential way to overcome this issue is to use more geometrical information coming from the 3D object to compute directly the ICR on the object surface, without relying so much on the wrench space structure; this is currently an ongoing work.

VI. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 287787, project SMERobotics.

REFERENCES

[1] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Ratón, Florida: CRC Press, 1994.
[2] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3D object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.
[3] V. Nguyen, "Constructing force-closure grasps," *Int. J. Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.
[4] J. Ponce and B. Faverjon, "On computing three-finger force-closure grasps of polygonal objects," *IEEE Trans. Robotics and Automation*, vol. 11, no. 6, pp. 868–881, 1995.
[5] J. Ponce, S. Sullivan, A. Sudsang, J. Boissonat, and J. Merlet, "On computing four-finger equilibrium and force-closure grasps of polyhedral objects," *Int. J. Robotics Research*, vol. 16, no. 1, pp. 11–35, 1997.
[6] N. Pollard, "Closure and quality equivalence for efficient synthesis of grasps from examples," *Int. J. Robotics Research*, vol. 23, no. 6, pp. 595–614, 2004.
[7] M. A. Roa and R. Suarez, "Computation of independent contact regions for grasping 3-D objects," *IEEE Trans. Robotics*, vol. 25, no. 4, pp. 839–850, 2009.
[8] R. Krug, D. Dimitrov, K. Charusta, and B. Iliev, "On the efficient computation of independent contact regions for force closure grasps," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 586–591.
[9] C. Rosales, L. Ros, J. Porta, and R. Suarez, "Synthesizing grasp configurations with specified contact regions," *Int. J. Robotics Research*, vol. 30, no. 4, pp. 431–443, 2011.
[10] M. A. Roa, K. Hertkorn, C. Borst, and G. Hirzinger, "Reachable independent contact regions for precision grasps," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 5337–5343.
[11] K. Hertkorn, M. A. Roa, M. Brucker, P. Kremer, and C. Borst, "Virtual reality support for teleoperation using online grasp planning," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013.
[12] M. A. Roa and R. Suarez, "Influence of contact types and uncertainties in the computation of independent contact regions," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 3317–3323.
[13] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1992, pp. 2290–2295.
[14] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996.