

# Regrasp Planning in the Grasp Space Using Independent Regions

Maximo A. Roa and Raul Suarez

**Abstract**—This paper presents an approach for quasi-static regrasp planning using  $n$  fingers, taking advantage of a method that quickly explores the grasp space for discrete objects. The approach relies on a sampling method, which provides samples of force-closure or non force-closure grasps used to compute regions of the graspable or non-graspable space, respectively. The regrasp contact points generated assure that a force-closure grasp is always possible when performing the regrasp motions. Application examples are included to show the relevance of the results.

**Index Terms**—Regrasp planning, independent contact regions, non-graspable regions.

## I. INTRODUCTION

The manipulation of objects by multifingered hands has received considerable attention in the last years. The manipulation problem appears when an object grasped by an end-effector needs a grasp change during the execution of the task. This problem has been tackled using two different approaches: finger gaiting and regrasping. Finger gaiting involves the relocation of one of the fingers on the surface of the object while keeping the grasp with the remaining fingers [1]. The change of a grasp from  $n$  fingers to  $n - 1$  fingers involves a change in the problem conditions, as the number of degrees of freedom of the hand-object system increases when one contact is lost.

The regrasping approach (or multi-fingered manipulation) solves the manipulation problem by using all the available fingers on the hand; the positions of the fingers can only be changed by rolling or sliding them along the surface of the object [2]. The finger gaiting and regrasping movements can also be combined to plan a desired manipulation of the object; for instance, the manipulation problem may be represented as a *switching graph*, where each node represents a grasp, and the finger gaiting planning is reduced to a graph search problem [3]–[5].

The analysis of a grasp that fully restrains the object to resist the influence of external disturbances is mainly based on the properties of form or force-closure, depending on whether the position of the contacts or the forces applied by the fingers ensure the object immobility [6]. The requirement of maintaining the force-closure (FC) property of the grasp while performing the manipulation complicates the planning problem [3]. To provide robustness to the grasp to finger

positioning errors, the concept of independent contact regions (ICRs) on the object boundary was introduced [7]. The positioning of a finger in each ICR assures an FC grasp, independently of the exact position of each finger. The computation of ICRs has been solved for 2D polygonal [8] and 3D polyhedral objects [9], as well as for objects of arbitrary shape described by a mesh with a large number of points, with frictional and frictionless contacts for 2D [10] and 3D [11] objects. A related concept is the notion of non-graspable regions (NGRs), defined such that a finger contact in each NGR always produce a non-FC grasp, independently of the exact position of each finger [12].

The problem tackled in this paper is the search of trajectories for the fingertips on an object surface, in order to change from an initial FC grasp to a final desired one without losing the FC condition. The problem is solved by finding a path between the initial and final grasps in a grasp space representing all the possible grasps of the object. The grasp space is explored by taking samples, evaluating whether each sample represents a FC or non-FC grasp, and respectively computing the corresponding ICRs or sets of NGRs (hereafter called NGRHs). This allows a quick exploration of the grasp space. The ICRs and the NGRHs are regions of the grasp space that are represented as nodes of a regrasp graph, which is then searched for a solution path. The detailed algorithms used to compute the ICRs and NGRHs were presented in a previous work [12], as well as a comparative study of some sampling approaches to explore the grasp space [13]. Those results are used here to find the solution to the regrasping problem.

The approach used in this work focuses only on the geometry of the object and on the FC property to find the trajectories for the fingertips on the object surface, i.e. it is object-centered. The hand kinematics is not considered (some other works include it in the grasp computation, e.g. [14]). It is assumed that the manipulation is performed at low velocities, therefore the interaction forces are dominant compared to the inertial forces, and the manipulation can be considered quasi-statical. The basic quasi-static manipulation motions are rolling and sliding of the fingers on the object surface [15].

The rest of the paper is organized as follows. Section II provides the required background on FC grasps and grasp space, and explains the computation of ICRs and NGRHs. Section III describes the approach proposed to plan a regrasp movement on a discrete object. Section IV shows two examples to illustrate the approach, and, finally, Section V presents the conclusions of the work.

This work was partially supported by the Spanish Government through the projects DPI2007-63665 and DPI2008-02448.

M. Roa is with the Institute of Industrial and Control Engineering (IOC), Technical University of Catalonia (UPC). [maximo.roa@upc.edu](mailto:maximo.roa@upc.edu)

R. Suarez is with the IOC - UPC, Barcelona, Spain. [raul.suarez@upc.edu](mailto:raul.suarez@upc.edu)

## II. INDEPENDENT CONTACT REGIONS AND SETS OF NON-GRASPABLE REGIONS

This subsection presents the assumptions and basic notions to deal with the problem of regrasp planning, and summarizes the procedures presented in [12] to compute the independent contact regions (ICRs) and sets of non-graspable regions (NGRHs) for an FC and non-FC grasp, respectively.

### A. Assumptions

The following assumptions are considered in this work. There is a frictional punctual contact between each finger and the object, with friction being modeled according to Coulomb's law. The object surface is discretized with a large enough set  $\Omega$  of points  $\mathbf{p}_i$ , whose positions are described by one or two parameters  $u$  for 2D or 3D objects, respectively. The normal direction  $\hat{\mathbf{n}}_i$  pointing toward the interior of the object at  $\mathbf{p}_i$  is known. It is assumed that the surface is smooth, i.e. the normal vectors have small variations between two sampled points. Each point is connected with a set of neighboring points forming a mesh; the number of neighbors is irrelevant and therefore different types of mesh are valid.

### B. Grasp space

An  $n$ -finger grasp  $G$  is described by the set of parameters  $u_i$  that determine the positions of the fingers on the grasped object surface, i.e.  $G = \{u_1, \dots, u_p\}$ , with  $p = n$  for 2D objects and  $p = 2n$  for 3D objects. The  $p$ -dimensional space representing the position of the possible contact points defined by  $u_1, \dots, u_p$  is called the grasp space  $\mathcal{G}$  (also known as grasp configuration space or contact space [16]).  $\mathcal{G}$  has some symmetries, as any grasp  $G = \{u_1, \dots, u_p\}$  accounts for  $K$  different grasps, with  $K = n!$  being the total number of possible permutations of the fingers on the object while keeping the same contact points. This symmetry is used to ease the generation of the grasp space.

Fig. 1 shows an example of  $\mathcal{G}$  for an ellipse discretized with 64 points, and grasped with 3 frictional fingers; each point of  $\mathcal{G}$  represents 3 contact points on the ellipse. The grasp space  $\mathcal{G}$  contains  $64^3 = 262,144$  grasps, with 12.1% of FC grasps and 87.9% of non-FC grasps, as shown in Fig. 1b with dark and light colors, respectively.

### C. Force-closure condition

A unitary force  $\mathbf{f}_i$  applied on the object at  $\mathbf{p}_i$  along the surface normal direction generates a torque  $\boldsymbol{\tau}_i = \mathbf{p}_i \times \mathbf{f}_i$ ;  $\mathbf{f}_i$  and  $\boldsymbol{\tau}_i$  are grouped together in a wrench vector  $\boldsymbol{\omega}_i = (\mathbf{f}_i, \boldsymbol{\tau}_i)^T$ . The resultant wrench applied on the object can be expressed as a positive linear combination of wrenches applied at the contact points, which are grouped in a wrench set  $W$ . For frictionless grasps, the grasp forces can only be applied in the direction normal to the object boundary, thus  $W = \{\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_n\}$ . For frictional grasps, the grasp forces lie inside a friction cone that can be linearized

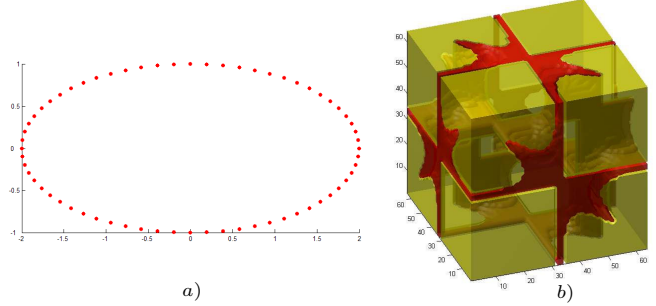


Fig. 1. Grasp space for a 2D object with 3 frictional contacts: a) Discretized ellipse, b) Grasp space.

with an  $m$ -side polyhedral convex cone, then the grasping force  $\tilde{\mathbf{f}}_i$  at the contact point  $\mathbf{p}_i$  can be expressed as

$$\tilde{\mathbf{f}}_i = \sum_{j=1}^m \alpha_{ij} \mathbf{s}_{ij}, \quad \alpha_{ij} \geq 0 \quad (1)$$

with  $\mathbf{s}_{ij}$  being the unitary vector along the  $j$ -th edge of the convex cone. The wrench produced by the force  $\tilde{\mathbf{f}}_i$  is

$$\tilde{\boldsymbol{\omega}}_i = \sum_{j=1}^m \alpha_{ij} \boldsymbol{\omega}_{ij}, \quad \boldsymbol{\omega}_{ij} = \begin{pmatrix} \mathbf{s}_{ij} \\ \mathbf{p}_i \times \mathbf{s}_{ij} \end{pmatrix} \quad (2)$$

where  $\boldsymbol{\omega}_{ij}$  is called a primitive contact wrench. Therefore, for frictional grasps  $W = \{\boldsymbol{\omega}_{11}, \dots, \boldsymbol{\omega}_{1m}, \dots, \boldsymbol{\omega}_{n1}, \dots, \boldsymbol{\omega}_{nm}\}$ .

A necessary and sufficient condition for the existence of an FC grasp is that the origin of the wrench space lies strictly inside the convex hull of  $W$ ,  $CH(W)$  [17]. This condition is applied in this work using the following lemma [12].

*Lemma 1:* Let  $G$  be a grasp with an associate set of wrenches  $W$ ,  $\mathcal{I}$  be the set of strictly interior points of  $CH(W)$ , and  $H_k$  be a supporting hyperplane of  $CH(W)$  (i.e. a hyperplane containing the facet  $k$  of  $CH(W)$ ). The origin  $O$  of the wrench space satisfies  $O \in \mathcal{I}$  if and only if  $\forall k$  any point  $P \in \mathcal{I}$  and  $O$  lie in the same half-space defined by  $H_k$ .

In this paper Lemma 1 is used selecting  $P$  as the centroid of the primitive contact wrenches, which is always an interior point of  $CH(W)$ . Then, the test used to verify the FC property for the grasp  $G$  checks whether the centroid  $P$  and the origin  $O$  lie on the same side of  $H_k \forall k$ .

### D. Independent contact regions

Let  $F_k$  denote a facet of  $CH(W)$  that contains at least one primitive wrench for a particular grasp point  $\mathbf{p}_i$ . The proposed approach builds hyperplanes  $H_k''$  parallel to each facet  $F_k$  and containing the origin  $O$  of the wrench space. These hyperplanes define the search zone  $S_i$ , containing the wrenches associated with physical points that belong to the  $ICR_i$  corresponding to  $\mathbf{p}_i$ .  $S_i$  is the intersection of the open half-spaces  $H_k''^+$  that contain the point  $\mathbf{p}_i$ .  $ICR_i$  is determined by the set of neighbor points of  $\mathbf{p}_i$  such that at least one of its primitive wrenches  $\boldsymbol{\omega}_{ij}$  falls into  $S_i$ . The algorithm is:

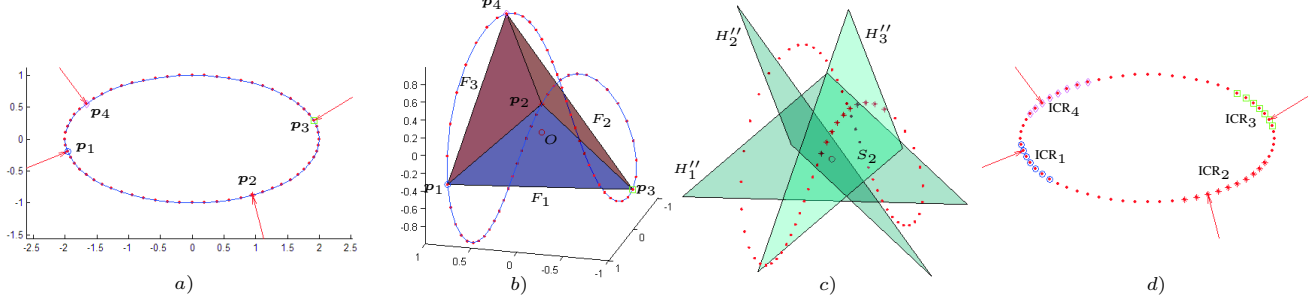


Fig. 2. Search of the ICRs for a discretized ellipse: a) Initial FC grasp, b) FC grasp in the wrench space, c) Search of ICR<sub>2</sub> for the point  $p_2$ , d) ICRs on the ellipse.

*Algorithm 1: Search of the independent contact regions* [12]

- 1) Find a starting FC grasp  $G = \{u_1, \dots, u_p\}$  and compute its corresponding wrench set  $W$
- 2) Compute the convex hull  $CH(W)$
- 3) For  $i = 1$  to  $n$  (i.e. for each contact point  $p_i \in \mathcal{C}$ ) do
  - a) For each facet  $F_k$  of  $CH(W)$  that has at least one vertex  $\omega_{ij}$ , build the hyperplane  $H''_k$  parallel to  $F_k$  and containing the origin  $O$ . Let  $H''_k^+$  be the open half-space such that  $\omega_{ij} \in H''_k^+$ , and let  $S_i$  be the search region such that  $S_i = \bigcap_k H''_k^+$
  - b) Initialize  $ICR_i = \{p_i\}$
  - c) Label  $p_i$  as open
  - d) While there are open points  $p_h \in ICR_i$  do
    - i) For all the neighboring points  $p_s$  of  $p_h$  do
      - If  $\exists j$  such that  $\omega_{sj} \in S_i$  then
      - $ICR_i = ICR_i \cup \{p_s\}$
      - Label  $p_s$  as open
    - ii) Label  $p_h$  as closed
- 4) Return the ICRs

Fig. 2 illustrates the search of the ICRs for a 2D object (in order to obtain 3D visualizations) using a 4-finger frictionless grasp of an ellipse discretized with 64 points. The initial FC grasp is shown on the ellipse and in the wrench space (Fig. 2a and 2b); continuous lines join the neighbor points. The computation of the ICR for the grasp point  $p_2$  is illustrated in Fig. 2c; three hyperplanes  $H''_k$  determine the search zone  $S_2$ , and the wrenches corresponding to the neighboring points of  $p_2$  that fall in  $S_2$  are depicted as stars. Fig 2d shows the ICRs for the 4 grasp points; 3,920 different FC grasps can be obtained from the possible combinations of finger positions inside the ICRs.

### E. Sets of non-graspable regions

The computation of the sets of non-graspable regions (NGRHs) starts with a non-FC grasp. First, the hyperplanes  $H''_k$ , parallel to each facet  $F_k$  and containing the origin  $O$  of the wrench space, are built. Then, the subset  $T$  of hyperplanes  $H''_k$  that completely leave  $CH(W)$  in the same open half-space are determined (i.e. if a plane  $H''_k$  intersects with  $CH(W)$  then it does not belong to  $T$ ). Every hyperplane in  $T$  defines a search zone  $S_T$  (open half-space) that fully contains  $CH(W)$ . The  $NGRH_i$  is determined by the set of

neighboring points of  $p_i$  such that all its primitive wrenches lie in  $S_T$ . The algorithm is:

*Algorithm 2: Search of the sets of non-graspable regions* [12]

- 1) Find a starting non-FC grasp  $G = \{u_1, \dots, u_p\}$  and compute its corresponding wrench set  $W$
- 2) Compute  $CH(W)$
- 3) For each facet  $F_k$  of  $CH(W)$  build the hyperplane  $H''_k$  parallel to  $F_k$  and containing the origin  $O$ . Let  $T$  be the set of hyperplanes  $T = \{H''_t = H''_k \mid CH(W) \subset H''_k^+\}$  (i.e. a hyperplane  $H''_k$  belongs to  $T$  if and only if  $\omega_{i1} \wedge \dots \wedge \omega_{im} \in H''_k^+$  for every  $p_i$ ).
- 4) For  $j = 1$  to  $t$  (i.e. for each hyperplane  $H''_j \in T$ ) let  $S_T|_{H''_j = H''_j^+}$ 
  - For  $i = 1$  to  $n$  (i.e. for each contact point  $p_i \in \mathcal{C}$ ) do
    - a) Initialize  $NGRH_i = \{p_i\}$
    - b) Label  $p_i$  as open
    - c) While there are open points  $p_h \in NGRH_i$  do
      - i) For all the neighboring points  $p_s$  of  $p_h$  do
        - If  $\omega_{s1} \wedge \dots \wedge \omega_{sm} \in S_T|_{H''_j}$  then
        - $NGRH_i = NGRH_i \cup \{p_s\}$
        - Label  $p_s$  as open
      - ii) Label  $p_h$  as closed

Return the  $NGRHs|_{H''_j}$

Figure 3 shows a non-FC grasp on the ellipse and in the wrench space; two hyperplanes  $H''_1$  and  $H''_2$  belong to the set  $T$  and therefore are considered to compute the NGRHs. Fig. 4a and 4c show the two hyperplanes separately and the corresponding  $NGRHs|_{H''_1}$  and  $NGRHs|_{H''_2}$  in the wrench space, and Fig. 4b and 4d show them on the ellipse. The  $NGRHs|_{H''_1}$  and  $NGRHs|_{H''_2}$  allow 44,100 and 2,313,441 different non-FC grasps, respectively.

### III. REGRASP PLANNING

The regrasp planning problem is formulated as follows: given an initial and a final FC grasps,  $G_i$  and  $G_f$  respectively, find a trajectory for each finger contact on the object surface that allows the grasp change while keeping the FC property (i.e. ensuring the resistance to any external disturbance appeared during the regrasp process). The sequence of movements corresponds to a path between the points  $G_i$  and  $G_f$  in the grasp space  $\mathcal{G}$  such that all the points in the path must be FC grasps.

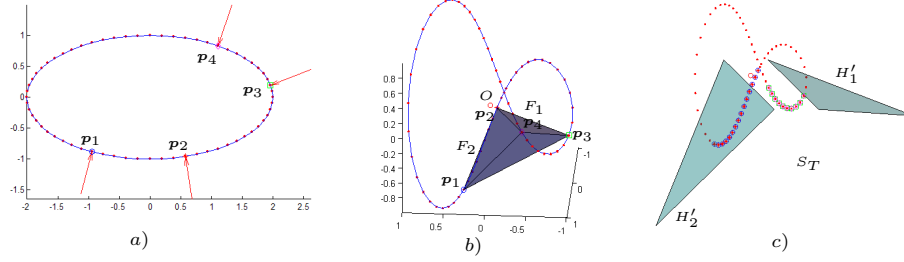


Fig. 3. Example of a non-FC grasp for a discretized ellipse: a) Initial non-FC grasp, b) Non-FC grasp in the wrench space, c) Hyperplanes  $H'_k$  belonging to the set  $T$ .

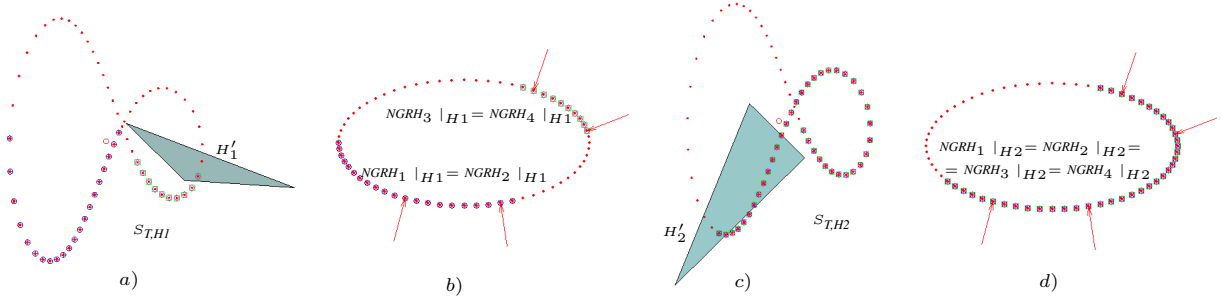


Fig. 4. Search of the sets of non-graspable regions for the previous example: a) Hyperplane  $H'_1$  and NGRHs in the wrench space, b) Sets  $NGRH|_{H_1}$  on the ellipse, c) Hyperplane  $H'_2$  and NGRHs in the wrench space, d) Sets  $NGRH|_{H_2}$  on the ellipse.

The regrasp algorithm uses the concepts of ICRs and NGRHs to compute the path in the grasp space. The ICRs and NGRHs define an axis-aligned box BI or BN in the grasp space, respectively (Fig. 5); the projection of the box along each of the axis corresponds to the ICR or NGRH for each one of the fingers. An auxiliary regrasp graph (hereafter called RG) is required to represent each BI box as a node, and a pair of contiguous BIs in the grasp space is represented as a pair of nodes with an arc between them. The regrasp algorithm first computes the BIs for  $G_i$  and  $G_f$ . Then, the algorithm takes a sample grasp from  $\mathcal{G}$ , identifies whether it is FC or not, and builds the corresponding box around it. If it is an FC grasp, then the computed box BI (defined by the ICRs) is added to RG, and the contiguity relations for the new BI are tested, i.e. new arcs are added to RG between the nodes representing BIs that intersect each other. If the sample grasp is a non-FC grasp, then all the possible grasps included in the box BN (defined by the NGRHs) are labeled as non-FC grasps, and are discarded from the following sampling steps. The iterative procedure goes on until a continuous path is obtained in the regrasp graph (or, equivalently, in the grasp space), or until all the grasp space has been explored and no path has been found. The algorithm is

*Algorithm 3: Regrasp planning*

- 1) For the initial and final grasps,  $G_i$  and  $G_f$  respectively:
  - a) Compute the boxes  $BI_i$  and  $BI_f$  (using Algorithm 1)
  - b) Label all the possible grasps inside the BIs as FC grasps
  - c) Add the boxes BI as nodes to the regrasp graph RG
- 2) Get a sample grasp  $G_s$
- 3) If  $G_s$  is FC

- a) If  $G_s$  has been already labeled as an FC grasp, go to Step 2
  - b) Compute the box  $BI_s$  (using Algorithm 1)
  - c) Label all the possible grasps inside  $BI_s$  as FC grasps
  - d) Add the new  $BI_s$  as a node in RG
  - e) Determine the contiguity relations between the new  $BI_s$  and the existing BIs in RG
- Else (i.e. if  $G_s$  is non-FC)
- a) If  $G_s$  has been already labeled as a non-FC grasp, go to Step 2
  - b) Compute the box  $BN_s$  (using Algorithm 2)
  - c) Label all the possible grasps inside  $BN_s$  as non-FC grasps
- 4) If there is a path in RG between the boxes  $BI_i$  and  $BI_f$ , compute the regrasp trajectory and return it
  - Else, go to Step 2

Figure 5 illustrates the algorithm for a hypothetical 2-dimensional grasp space. It is considered that the order of parameters in the grasps  $G_i$  and  $G_f$  respects a predefined assignment of fingers, i.e. the first parameters,  $u_{1i}$  and  $u_{1f}$ , describe the position of finger 1 in the initial and final grasp, respectively; then, the regrasp planner must provide a sequence of positions to move each finger  $k$  from the position  $u_{ki}$  to  $u_{kf}$ . The sampling method used in Step 2 is based on a structured grid that identifies each cell of  $\mathcal{G}$  with a unique numerical code [13]. The sample selection follows a deterministic sequence that assures the completeness of the method (a complete deterministic sequence covers the whole grasp space). Even if it is not explicitly stated in Algorithm 3, Step 2 returns “no solution” if all the grasp space has been explored and no path has been found.

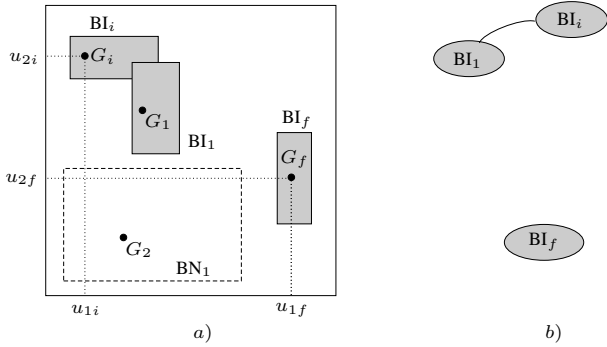


Fig. 5. Regrasp planning: a) Hypothetical 2-dimensional grasp space with the initial and final grasps and BIs, one sampled FC grasp and one non-FC grasp, b) Regrasp graph RG with the contiguity relations between the nodes that represent the BIs.

Step 4 checks whether there is a path between the initial and final grasp; this is performed using a Dijkstra algorithm applied to the regrasp graph RG. For a quicker convergence of the algorithm (to a solution or to completely cover  $\mathcal{G}$  and decide that there is no solution at all), Step 4 could be executed every certain number of generated samples. When there is a path between the initial and final grasps, obtained as a sequence of ICRs in RG, the regrasp trajectory must be computed in  $\mathcal{G}$ ; different criteria can be used to compute such trajectory (for instance, minimizing the number of grasp changes). The regrasp trajectory that this planner provides is based on one-at-a-time movements of the fingers, i.e. the trajectory of the regrasp in the grasp space follows the direction of the axis.

As it was stated before, the ICRs and NGRHs define an axis-aligned box BI or BN in the grasp space. Each of them is stored by using  $2p$  parameters, representing the lower and upper limit of the correspondent box in each axis of  $\mathcal{G}$ . Note that due to the symmetries of  $\mathcal{G}$ , each BI or BN actually corresponds to  $K = n!$  axis-aligned boxes in  $\mathcal{G}$  (the total number of possible permutations of the fingers on the object while keeping the same contact points). Fig. 6 shows an example of BIs and BNs obtained for a 3-finger frictional grasp of a discrete ellipse. The initial FC and non-FC grasps are also shown, as well as the corresponding boxes in the grasp space.

#### IV. EXAMPLES

To illustrate the proposed approach, the algorithms were implemented in Matlab on a Pentium IV 3.2 GHz PC. The following examples show the regrasp planning process for 3-finger frictional grasps on 2D objects. 2D examples were selected for ease of visualization, as the corresponding grasp space is 3-dimensional and can be graphically represented.

##### A. Example 1

The first example uses an ellipse discretized with 64 points (Fig. 1a); the FC and non-FC grasp space is shown in Fig. 1b with dark and light colors, respectively. The FC grasp space explored while searching for the regrasp sequence is shown

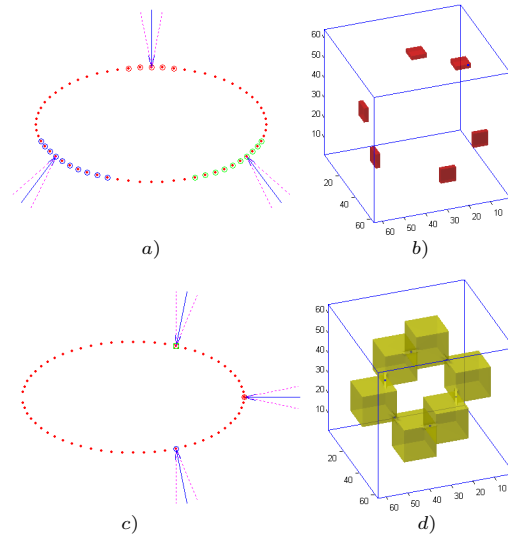


Fig. 6. ICRs and NGRHs for a discretized ellipse: a) ICRs for an initial FC grasp, b) Corresponding boxes BI in the grasp space, c) Initial non-FC grasp, d) Corresponding boxes BN in the grasp space.

TABLE I  
RESULTS FOR THE REGRASP COMPUTATION IN EXAMPLE 1

Parameter	Regrasp computation	Total grasp space
time [s]	20.5	2,619
ICRs computed	102	564
% of the FC space	66.7	100
NGRHs computed	72	350
% of the non-FC space	95.7	100

in Fig. 7a. Figure 7b shows the regrasp path inside the contiguous BIs that connect the initial and final grasp. As a comparison, Fig. 7c shows the whole FC grasp space.

In 10 trials of regrasp computations between the same initial and final grasp, the averaged total time elapsed to get the regrasp sequence is 20.5 s, and required 102 evaluations of ICRs and 72 evaluations of NGRHs. Table I compares these results to the averaged results for the total exploration of the whole grasp space using the deterministic sampling process [13]. Note that the regrasp computation provides a feasible trajectory in a very short time when compared to the time required for the total exploration of the grasp space, because the deterministic exploration provides a fast exploration of a large portion of the grasp space, but the more space it covers the more slowly the percentage of the grasp space covered increases.

##### B. Example 2

The second example uses an object defined by a closed parametric curve presented in [18] and discretized with 128 points (Fig. 9a). The resulting grasp space is shown in Fig. 9b; it contains  $128^3 = 2,097,152$  grasps, with 12.2% and 87.8% of FC and non-FC grasps, respectively.

Figure 10 shows the initial and final grasp used for a regrasp planning. Figure 11 shows an example of regrasp computation on this object. Fig. 11a show the FC grasp space explored while searching for the regrasp sequence between

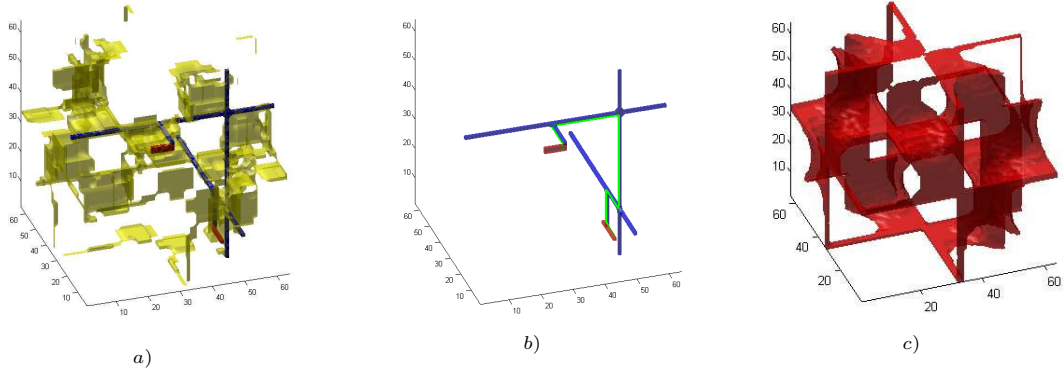


Fig. 7. Regrasp planning for Example 1: a) FC space explored while searching the regrasp sequence; b) Contiguous BIs that provide the regrasp path between the initial and final grasp; c) Total FC space for the example.

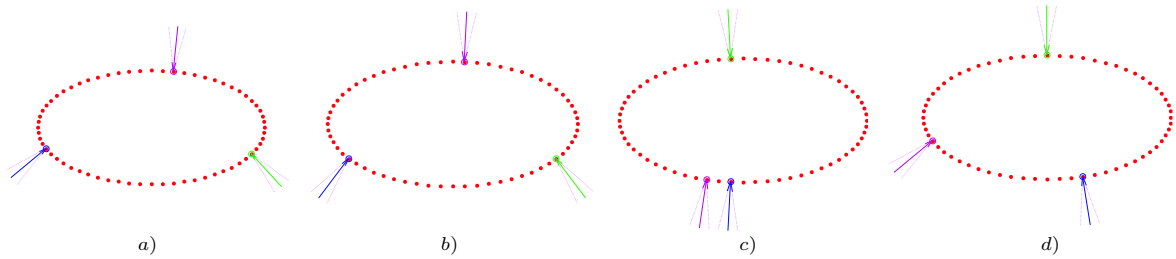


Fig. 8. Sequence of grasps for Example 1: a) Initial grasp  $G_i$ ; b) and c) Intermediate grasps; d) Final grasp  $G_f$ .

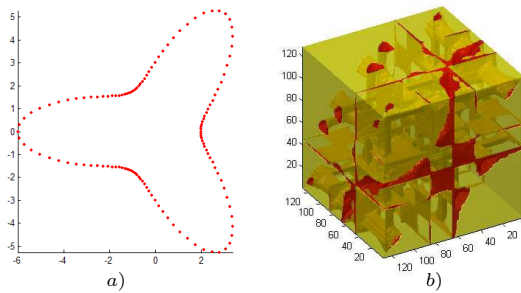


Fig. 9. Example 2: a) Discrete object, b) Grasp space.

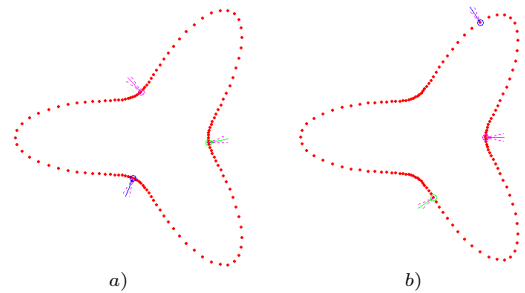


Fig. 10. Example 2: a) Initial grasp, b) Final grasp.

TABLE II

RESULTS FOR THE REGRASP COMPUTATION IN EXAMPLE 2

Parameter	Regrasp computation	Total grasp space
time [s]	840	172,740
ICRs computed	653	4,111
% of the FC space	67.5	100
NGRHs computed	669	2,649
% of the non-FC space	94.1	100

the initial and final grasp. Figure 11b shows the regrasp path inside the contiguous BIs that connect the initial and final grasp. Finally, Fig. 11c shows the whole FC grasp space for this object. Table II compares the results for this example with the averaged results for the total exploration of the grasp space using the deterministic sampling process.

## V. CONCLUSIONS

This paper has presented an efficient approach to generate a regrasp trajectory in the grasp space, valid for 2D and

3D discrete objects and for any number of fingers. The proposed method is based on the concepts of independent contact regions (ICRs) and sets of non-graspable regions (NGRHs). The approach takes samples of the grasp space; if a sample is an FC grasp then the ICRs are computed, if it is a non-FC grasp then the set of NGRHs is computed. The ICRs and NGRHs define an axis-aligned box BI or BN in the grasp space, respectively. Each box BI or BN provides a number of additional FC or non-FC grasps, and therefore with a low number of samples a large portion of the grasp space is covered. The search of a regrasp path is converted into a graph search in a regrasp graph, that keeps trace of the contiguity relations between different BIs in the grasp space.

The algorithms presented in the paper have been implemented and some application examples are given. The procedures are fully valid for 3D objects with high-dimensional grasp spaces, however, the application to 3D objects requires

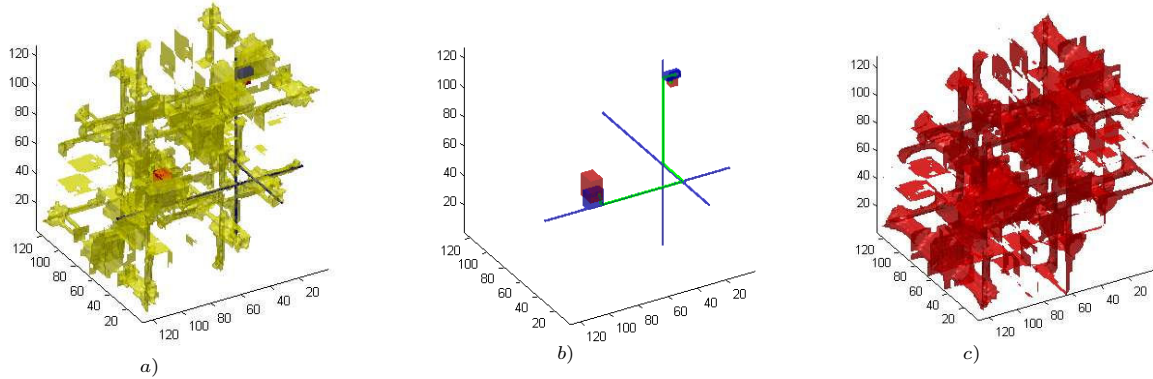


Fig. 11. Regrasp planning for Example 2: a) FC space explored while searching the regrasp sequence; b) Contiguous BIs that provide the regrasp path between the initial and final grasp; c) Total FC space for the example.

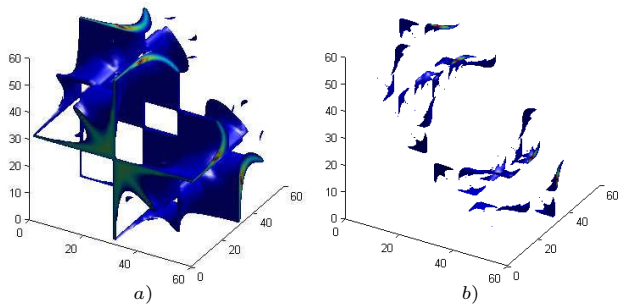


Fig. 12. Grasp quality in the grasp space for a discretized ellipse: a)  $Q = 0.1$ , b)  $Q = 0.2$ .

an efficient way to save the data (the grasp space has high dimensionality, for instance it is 8-dimensional for a 4-finger frictional grasp on a 3D object); the development of an efficient storage method to speed up the application of the proposed algorithm to 3D discrete objects is an interesting line of future work.

Another extension of the proposed approach is to tackle the search of a regrasp path that assures a minimum grasp quality  $Q$ . The algorithms that compute the ICRs assuring a minimum grasp quality have already been developed [11], using as quality measure the largest perturbation wrench that the grasp can resist independently of the perturbation direction [17]. However, it should be noted that if the grasp quality increases there are more chances of not achieving a regrasp path between two given configurations. For instance, Fig. 12 shows the FC space with a grasp quality higher than a certain threshold  $Q$  (as a comparison, Fig. 7c shows the whole FC grasp space, with minimum quality  $Q \approx 0$ ). Note that for  $Q = 0.2$  there are several disconnected regions in the grasp space, and if the initial and final grasp belong to different regions, then there is no regrasp sequence that connects them while assuring a minimum quality higher than the defined threshold.

#### REFERENCES

[1] B. Goodwine and J. Burdick, "Motion planning for kinematic stratified systems with application to quasi-static legged locomotion and finger

gaiting," *IEEE Trans. Robotics and Automation*, vol. 18, no. 2, pp. 209–222, 2002.

[2] D. Montana, "The kinematics of multi-fingered manipulation," *IEEE Trans. Robotics and Automation*, vol. 11, no. 4, pp. 491–503, 1995.

[3] L. Han and J. Trinkle, "Object reorientation with finger gaiting," in *Proc. 2nd IMACS Int. Conf. Computational Engineering in Systems Applications*, 1998.

[4] T. Phoka, P. Pipattanasomporn, N. Niparman, and A. Sudsang, "Regrasp planning of four-fingered hand for a parallel grasp of a polygonal object," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2005, pp. 791–796.

[5] J. Xu, Y. Lou, and Z. Li, "Hybrid automaton: a better model of finger gaits," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 4628–4633.

[6] A. Bicchi, "On the closure properties of robotic grasping," *Int. J. Robotics Research*, vol. 14, no. 4, pp. 319–344, 1995.

[7] V. Nguyen, "Constructing force-closure grasps," *Int. J. Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.

[8] J. Cornella and R. Suarez, "Fast and flexible determination of force-closure independent regions to grasp polygonal objects," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2005, pp. 778–783.

[9] 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.

[10] J. Cornella and R. Suarez, "Determining independent grasp regions on 2D discrete objects," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 2936–2941.

[11] M. Roa and R. Suarez, "Independent contact regions for frictional grasps on 3D objects," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2008, pp. 1622–1627.

[12] M. Roa, R. Suarez, and J. Rosell, "Grasp space generation using sampling and computation of independent regions," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 2258–2263.

[13] —, "Influence of the sampling strategy on the incremental generation of the grasp space," in *Proc. 11th Int. Conf. on Climbing and Walking Robots - CLAWAR*, 2008.

[14] Y. Guan and H. Zhang, "Kinematical feasibility analysis of 3-D multifingered grasps," *IEEE Trans. Robotics and Automation*, vol. 19, no. 3, pp. 507–513, 2003.

[15] M. Cherif and K. Gupta, "Planning quasi-static fingertip manipulations for reconfiguring objects," *IEEE Trans. Robotics and Automation*, vol. 15, no. 5, pp. 837–848, 1999.

[16] 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.

[17] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1992, pp. 2290–2295.

[18] Y. Jia, "Computation on parametric curves with an application in grasping," *Int. J. Robotics Research*, vol. 23, no. 7-8, pp. 825–855, 2004.