Grasp Space Generation Using Sampling and Computation of Independent Regions

Máximo A. Roa, Raúl Suárez and Jan Rosell

***Abstract*— This paper presents the use of independent contact and non-graspable regions to generate the grasp space for 2D and 3D discrete objects. The grasp space is constructed via 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 method provides a reliable procedure for an efficient generation of the whole grasp space for n-finger grasps on discrete objects; two examples on 2D objects are provided to illustrate its performance. The approach has several applications in manipulation and regrasping of objects, as it provides a large number of force-closure and non force-closure grasps in a short time.**

***Index Terms*— Grasp space, independent contact regions, non-graspable regions.**

I. Introduction

Grasp planning searches for desirable locations of the fingers on the object surface, for instance, to achieve the object equilibrium, or to fully restrain the object to resist the influence of external disturbances. To immobilize the object the grasp must satisfy 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 [1]. These properties have been widely used in the synthesis of precision grasps (i.e. when only the fingertips touch the object) for 2D [2] [3] [4] and 3D objects [5] [6].

To provide robustness to the grasp in front of 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 a force-closure (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 [5] [9], as well as for objects of arbitrary shape described by a mesh with large number of points, for 2D [10] and 3D [11] [12] discrete objects, and with frictional and frictionless contacts. As an opposite concept to the ICRs, this paper introduces the non-graspable regions (NGRs) such that a finger contact in each NGR always produce a non-FC grasp, independently of the exact position of each finger

Most of the works above-mentioned focus on the synthesis of one grasp that optimizes a particular criterion. However, in applications such as manipulation and regrasp planning it is useful to know all the possible FC grasps or at least a large number of them, i.e. know the structure of the whole grasp space. Previous works have tackled the computation of all the n-finger FC grasps for 2D polygonal objects [2], and all the 3-finger FC grasps for 2D discretized objects [3]; to the best of the authors' knowledge, the generic computation of all the n-finger FC grasps for frictional and frictionless contacts in 2D and 3D discrete objects has not been tackled before. This paper presents a method to generate the grasp space for discrete objects using NGRs and ICRs, i.e. to compute all the FC and non-FC possible grasps.

The rest of the paper is organized as follows. Section II provides the required background on FC grasps and grasp space. Section III describes the approach proposed to generate the grasp space, and Section IV presents the algorithms to compute the ICRs and NGRs starting from a FC or non-FC sample grasp, respectively. Section V shows two examples to illustrate the approach, and, finally, Section VI summarizes the work and discusses some future applications.

II. Framework

A. Assumptions

In this work the following assumptions are considered. 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 in a large enough set Ω of points $\boldsymbol{p}_i$, whose positions are described by one or two parameters u for 2D or 3D objects, respectively, and the normal direction $\hat{\boldsymbol{n}}_i$ pointing toward the interior of the object at $\boldsymbol{p}_i$ is known. 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 and force-closure conditions

An n-finger grasp G is defined as the set of parameters u_i that fix 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 defined by $u_1, \dots, u_p$ is called the grasp space (also known as grasp configuration space or contact space) [9].

A unitary force $\boldsymbol{f}_i$ applied on the object at $\boldsymbol{p}_i$ along the surface normal direction generates a torque $\boldsymbol{\tau}_i = \boldsymbol{p}_i \times \boldsymbol{f}_i$; $\boldsymbol{f}_i$ and $\boldsymbol{\tau}_i$ are grouped together in a wrench vector $\boldsymbol{\omega}_i = (\boldsymbol{f}_i, \boldsymbol{\tau}_i)^T$. 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

This work was partially supported by the CICYT projects DPI2005-00112 and DPI2007-63665.

M. Roa is with the National University of Colombia. He is supported by Colfuturo, PCB-2006, and currently works at the Institute of Industrial and Control Engineering (IOC), Technical University of Catalonia (UPC). maximo.roa@upc.edu

R. Suarez and J. Rosell are with the IOC - UPC, Barcelona, Spain. raul.suarez@upc.edu, jan.rosell@upc.edu

$W = \{\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_n\}$. For frictional grasps, the grasp forces lie inside a friction cone that can be linearized with an m-side polyhedral convex cone, then the grasping force $\tilde{\boldsymbol{f}}_i$ at the contact point $\boldsymbol{p}_i$ can be expressed as

$$\tilde{\boldsymbol{f}}_i = \sum_{j=1}^{m} \alpha_{ij} \boldsymbol{s}_{ij},\ \alpha_{ij} \geq 0 \tag{1}$$

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

$$\tilde{\boldsymbol{\omega}}_{\boldsymbol{i}} = \sum_{j=1}^{m} \alpha_{ij} \boldsymbol{\omega}_{ij}\ ,\ \ \boldsymbol{\omega}_{ij} = \begin{pmatrix} \boldsymbol{s_{ij}} \\ \boldsymbol{p_i} \times \boldsymbol{s_{ij}} \end{pmatrix} \tag{2}$$

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 a FC grasp is that the origin of the wrench space lies strictly inside the convex hull of W, $CH(W)$ [13]. 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 i 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 $\forall i$.

III. Generation of the Grasp Space

The generation of the grasp space is based on the concepts of Independent Contact Regions (ICRs) and Non-Graspable Regions (NGRs). The ICRs and NGRs are regions such that positioning a finger anywhere inside each of them a FC or non-FC grasp will always be obtained, respectively. Basically, the algorithm takes a sample of the grasp space, identifies whether it is force-closure or not, and builds the corresponding region around it, labeling in this way a significant number of potential FC or non-FC grasps of the object. This action can be repeated until a useful portion of the grasp space is labeled (for instance for grasp or regrasp planning purposes) or simply until the whole grasp space is labeled. The algorithm is:

Algorithm 1: Exploration of the grasp space

1) Generate a sample grasp G
2) If G has not been previously labeled, test whether G is a FC grasp
   If G is FC
      Compute the ICRs
      Label G and every possible combination of grasps generated by choosing one point from each ICR as a FC grasp
   Else
      Compute the NGRs
      Label G and every possible combination of grasps generated by choosing one point from each NGR as a non-FC grasp
3) If the grasp space is not fully labeled yet (or a particular condition is not reached) then go to Step 1
   Else, Return the grasp space

The sampling method used to generate samples in Step 1 is based on a lattice structure where each cell of the grasp space is identified by an unique numerical code. The samples are randomly selected, and to assure the completeness of the method, the samples already chosen are eliminated from the sampling list for the next step.

IV. Computation of the Independent Contact and Non-Graspable Regions

A. Independent contact regions

This subsection summarizes the procedure presented in [12] to compute the independent contact regions (ICRs) for a FC grasp. Let F_k denote a facet of $CH(W)$ that contains at least one primitive wrench for a particular grasp point $\boldsymbol{p}_i$. 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 $\boldsymbol{p}_i$. S_i is the intersection of the open half-spaces ${H'_k}^+$ that contain the point $\boldsymbol{p}_i$. ICR$_i$ is determined by the set of neighbor points of $\boldsymbol{p}_i$ such that at least one of its primitive wrenches $\boldsymbol{\omega}_{ij}$ falls into S_i. The algorithm is:

Algorithm 2: Search of the independent contact regions

Initialize with a FC grasp $G = \{u_1, \dots, u_p\}$, and compute its corresponding wrench set W and the convex hull $CH(W)$.
For each contact point $\boldsymbol{p}_i$, $i = 1, \dots, n$, do

1) Build the hyperplanes H'_k parallel to each F_k and containing the origin O
2) Let $S_i = \bigcap_k {H'_k}^+$ with ${H'_k}^+$ the open half-space such that $\exists \boldsymbol{\omega}_{im} \in S_i$
3) Initialize $Z_i = \{\boldsymbol{p}_i\}$. Label the points in Z_i as open
4) While there are open points $\boldsymbol{p}_j \in Z_i$
   For every neighboring point $\boldsymbol{p}_s$ of $\boldsymbol{p}_j$
      If $\boldsymbol{\omega}_{s1} \vee \dots \vee \boldsymbol{\omega}_{sm} \in S_i$
         $Z_i = Z_i \cup \{\boldsymbol{p}_s\}$
         Label $\boldsymbol{p}_s$ as open
   Label $\boldsymbol{p}_j$ as closed
5) Return Z_i (i.e. the ICR$_i$ for the contact point $\boldsymbol{p}_i$)

Note that the algorithm is computationally very simple. In Step 1, the hyperplanes H'_k are computed for the corresponding facets F_k of $CH(W)$. Let H_k be the hyperplane containing the facet F_k, described as $\boldsymbol{e}_k \cdot \boldsymbol{x} = e_{0k}$. The hyperplane H'_k parallel to H_k and containing the origin is $\boldsymbol{e}_k \cdot \boldsymbol{x} = 0$, i.e. the parameters $\boldsymbol{e}_k$ of H'_k are the same as for H_k. Step 2 only identifies for every hyperplane the open half-space ${H'_k}^+$ that contains at least one of the primitive wrenches for the point $\boldsymbol{p}_i$, and forms the search zones S_i.

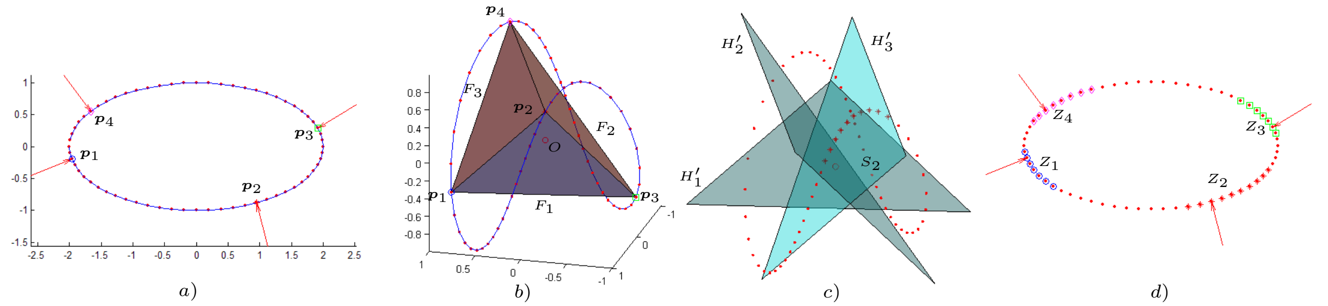

Fig. 1. Search of the ICRs for a discretized ellipse: a) Initial FC grasp, b) FC grasp in the wrench space, c) Search of ICR_2 for the point $\boldsymbol{p}_2$, d) ICRs on the ellipse.

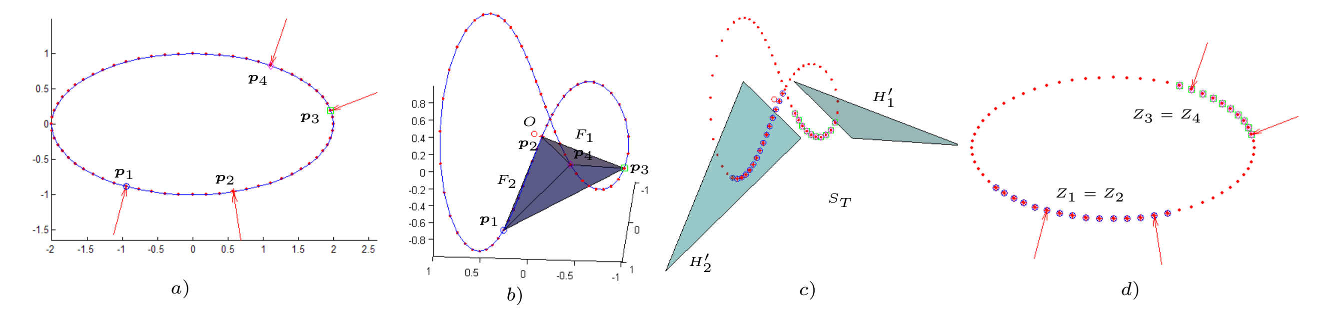

Fig. 2. Search of the non-graspable regions for a discretized ellipse: a) Initial non-FC grasp, b) Non-FC grasp in the wrench space, c) Search of the NGRs in the wrench space, d) NGRs on the ellipse.

Note that due to the geometrical construction any set of contact points that allow a wrench within each S_i always generates a FC grasp. Step 4 is the most complex step in the algorithm, it requires an iterative check of the primitive wrenches produced by each new neighboring point, which is done classifying the wrench with respect to the corresponding hyperplanes H_k. The number of points in each Z_i may be different, depending on factors such as the level of detail in the representation of the object surface and the smoothness of the surface, i.e. the rate of change in the normal vector around the contact location.

Fig. 1 illustrates the search of the ICRs. In order to obtain 3D visualizations, a simple case is presented: the search of ICRs for the 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. 1a and 1b); continuous lines join the neighbor points. The computation of the ICR for the grasp point $\boldsymbol{p}_2$ is illustrated in Fig. 1c; three hyperplanes H'_k determine the search zone S_2, and the wrenches corresponding to the neighboring points of $\boldsymbol{p}_2$ that fall in S_2 are depicted as stars. Fig 1d shows the ICRs for the 4 grasp points; 3920 different FC grasps can be obtained from the possible combinations of finger positions inside the ICRs.

B. Non-graspable regions

The computation of the non-graspable regions (NGRs) 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). The hyperplanes in T define a search zone S_T that fully contains $CH(W)$; S_T is the intersection of the open half-spaces H'^{+}_k that contain all the primitive wrenches corresponding to the point $\boldsymbol{p}_i$. The NGR_i is determined by the set of neighboring points of $\boldsymbol{p}_i$ such that all of its primitive wrenches lie in S_T. The algorithm is:

Algorithm 3: Search of the non-graspable regions

Initialize with a non-FC grasp $G = \{u_1, \ldots, u_p\}$, and compute its corresponding wrench set W and the convex hull $CH(W)$.

1) Build the hyperplanes H'_k parallel to each F_k and containing the origin O
2) Let $S_T = \bigcap_k H'^{+}_k$ with H'^{+}_k the open half-space such that $CH(W) \subset H'^{+}_k$ (i.e. $\boldsymbol{\omega}_{i1} \wedge \ldots \wedge \boldsymbol{\omega}_{im} \in H'^{+}_k$ for each $\boldsymbol{p}_i$)
3) Initialize $Z_i = \{\boldsymbol{p}_i\}$. Label the points in Z_i as open
4) For each contact point $\boldsymbol{p}_i$
   While there are open points $\boldsymbol{p}_j \in Z_i$
      For every neighboring point $\boldsymbol{p}_s$ of $\boldsymbol{p}_j$
         If $\boldsymbol{\omega}_{s1} \wedge \ldots \wedge \boldsymbol{\omega}_{sm} \in S_T$
            $Z_i = Z_i \cup \{\boldsymbol{p}_s\}$
            Label $\boldsymbol{p}_s$ as open
      Label $\boldsymbol{p}_j$ as closed
5) Return the sets Z_i (i.e. the NGRs for each contact point $\boldsymbol{p}_i$)

Again, the algorithm is computationally very simple. Note that as $O \notin S_T$, choosing every possible combination of one wrench from each Z_i always generates a $CH(W)$ that

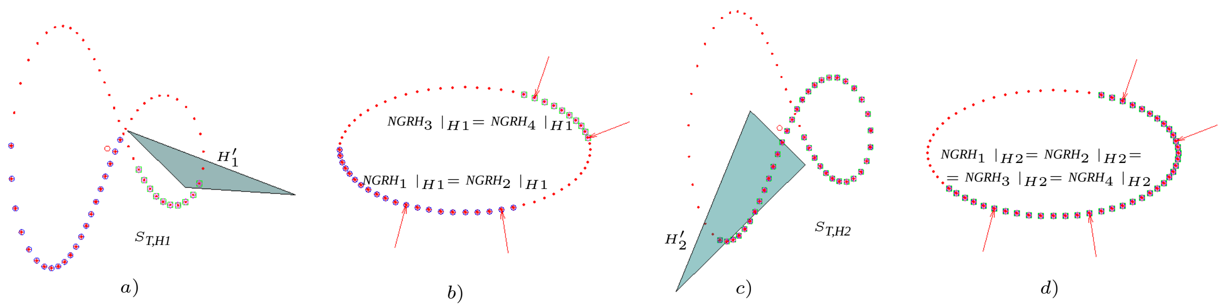

Fig. 3. Search of the non-graspable sets for the previous example: a) Hyperplane H_1' and NGRHs in the wrench space, b) Sets NGRH$|_{H1}$ on the ellipse, c) Hyperplane H_2' and NGRHs in the wrench space, d) Sets NGRH$|_{H2}$ on the ellipse.

does not contain the origin O of the wrench space, i.e. the corresponding grasp is non-FC. Fig. 2 illustrates the search of the NGRs for the 4-finger frictionless grasp of a discretized ellipse. The non-FC grasp is shown on the ellipse and in the wrench space (Fig. 2a and 2b). The computation of the NGRs is illustrated in Fig. 2c; two hyperplanes H_k' determine the search zone S_T, and the wrenches corresponding to the neighboring points of each $\boldsymbol{p}_i$ that fall in the search zone S_T are depicted. Fig 2d shows the NGRs for the 4 grasp points; note that $\text{NGR}_1 = \text{NGR}_2$ and $\text{NGR}_3 = \text{NGR}_4$. 22,500 different non-FC grasps can be obtained from the possible combinations of finger positions inside the NGRs.

Note that each hyperplane H_k' that fulfills the condition in Step 2 (i.e. $CH(W) \subset H_k'^{+}$) can generate its own set of NGRs, hereafter called NGRHs. Then, in order to maximize the number of non-FC grasps identified in each call to Algorithm 3, for each H_k' in T the search region can be redefined as $S_T = H_k'^{+}$, and the corresponding NGRHs are directly computed using Steps 3 and 4 in Algorithm 3. For instance, in the example in Fig. 2 two hyperplanes H_1' and H_2' are considered to compute the NGRs. Fig. 3a and 3c show the two hyperplanes separately and the corresponding NGRHs$|_{H1}$ and NGRHs$|_{H2}$ in the wrench space, and Fig. 3b and 3d show them on the ellipse. The NGRHs$|_{H1}$ and NGRHs$|_{H2}$ allow 44,100 and 2,313,441 different non-FC grasps, respectively. The equivalence of the NGRs in Fig. 2d with the NGRHs in Fig. 3 is given by $\text{NGR}_i = \text{NGRH}_i|_{H1} \cap \text{NGRH}_i|_{H2}$.

V. 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 generation of the grasp space for 3-finger frictional grasps on two 2D objects. 2D examples were selected for ease of visualization, as the corresponding grasp space is three-dimensional and can be graphically represented.

A. Example 1

The first example uses an ellipse discretized with 64 points (Fig. 4a). The grasp space contains $64^3 = 262,144$ grasps, with 12.1% of FC grasps and 87.9% of non-FC grasps, as shown in Fig. 4b with dark and light colors, respectively.

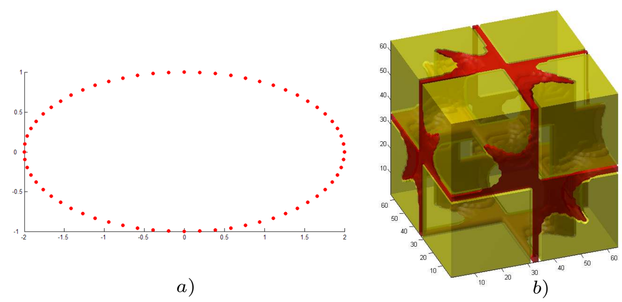

Fig. 4. Example 1: a) Ellipse, b) Grasp space.

The evolution in the generation of the grasp space using Algorithm 1 is presented in Fig. 5. The grasp space has symmetries, as any $G = \{u_1, u_2, u_3\}$ represents 6 equivalents grasps (the number of possible permutations of the 3 fingers on the ellipse while keeping the same 3 contact points); therefore, each set of ICRs or NGRs corresponds to six axis-aligned boxes in the grasp space.

Fig. 6 presents the percentage evolution in the coverage of the total grasp space; the results are the average of 20 different executions of the algorithm. With a low number of samples, the algorithm rapidly identifies a large portion of the grasp space, e.g. 82% of the grasp space has been already explored with just 100 samples, and 98% of the grasp space has been generated with 10^4 samples (3.8% of the total number of grasps). Fig. 7a presents the number of calls to Algorithms 2 and 3, i.e. the number of computations of ICRs and NGRs. Fig. 7b presents the time required to generate the grasp space.

B. Example 2

In the second example the object is defined by a closed parametric curve presented in [14] and discretized with 128 points (Fig. 8a). The resulting grasp space is shown in Fig. 8b; it contains $128^3 = 2,097,152$ grasps, with 12.2% and 87.8% of FC and non-FC grasps, respectively. Fig. 9 shows the evolution in the generation of the grasp space for different number of samples. Fig. 10 shows the percentage evolution in the coverage of the total grasp space (average of 20 trials). As for the previous example, for a low number of samples a large portion of the grasp space is covered, e.g.

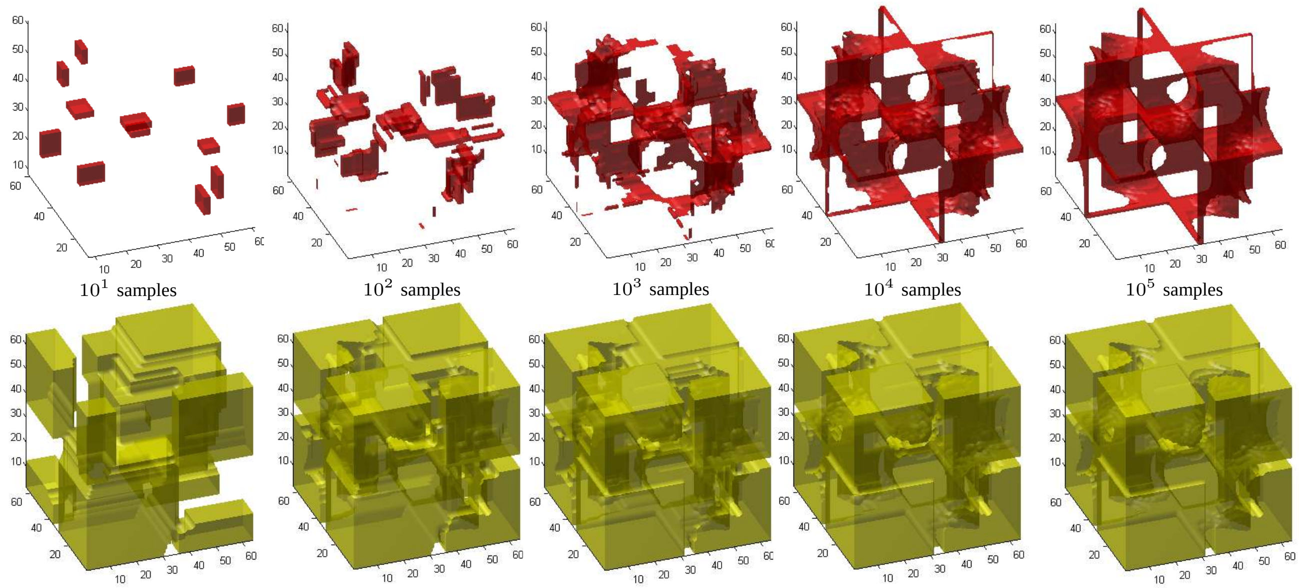

Fig. 5. Evolution in the generation of the grasp space for Example 1. Up: FC grasp space. Down: non-FC grasp space.

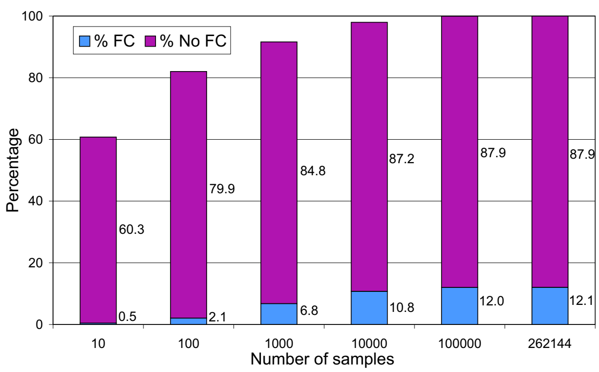

Fig. 6. Evolution of the grasp space generation for Example 1.

for 10,000 samples (0.48% of the total grasp space) 93.7% of the space has been generated, in ~ 1000 s.

VI. Conclusions

This paper has presented an efficient approach to generate the grasp space, valid for 2D and 3D discrete objects and for any number of fingers. The grasp space contains a large number of grasps, therefore, a brute-force exploration of the space would have a high computational cost. The proposed method is based on the concepts of independent contact regions (ICRs) and non-graspable regions (NGRs). The ICRs have been previously used in some works, but the concept of NGRs is introduced in this paper. The NGRs are defined as regions on the object boundary such that when a finger is positioned inside each of them, a non-FC grasp is always obtained, with independence of the exact position of each finger.

The proposed approach takes samples of the grasp space, if a sample is a FC grasp then the ICRs are computed, if it is a non-FC grasp then the NGRs are computed. Each ICR and

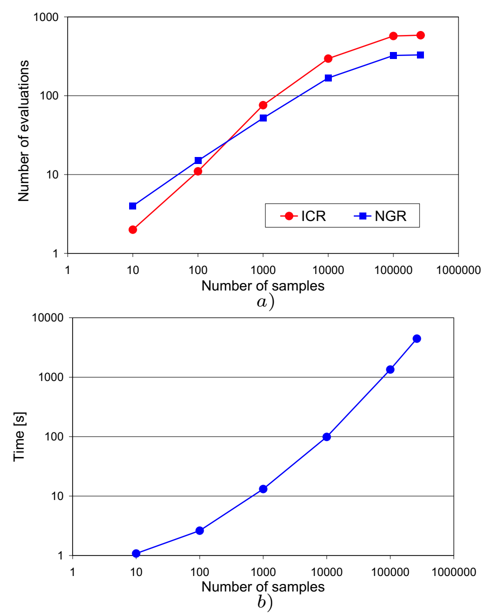

Fig. 7. Grasp space generation for Example 1: a) Computations of ICRs and NGRs, b) Computational time.

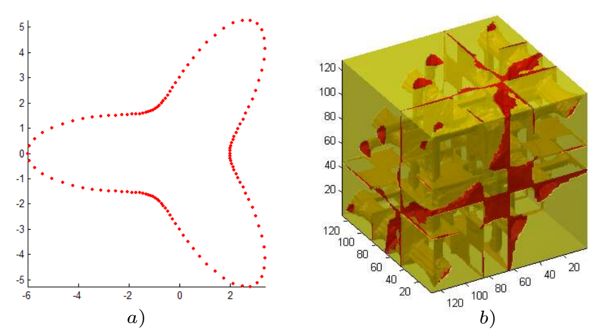

Fig. 8. Example 2: a) Discrete object, b) Grasp space.

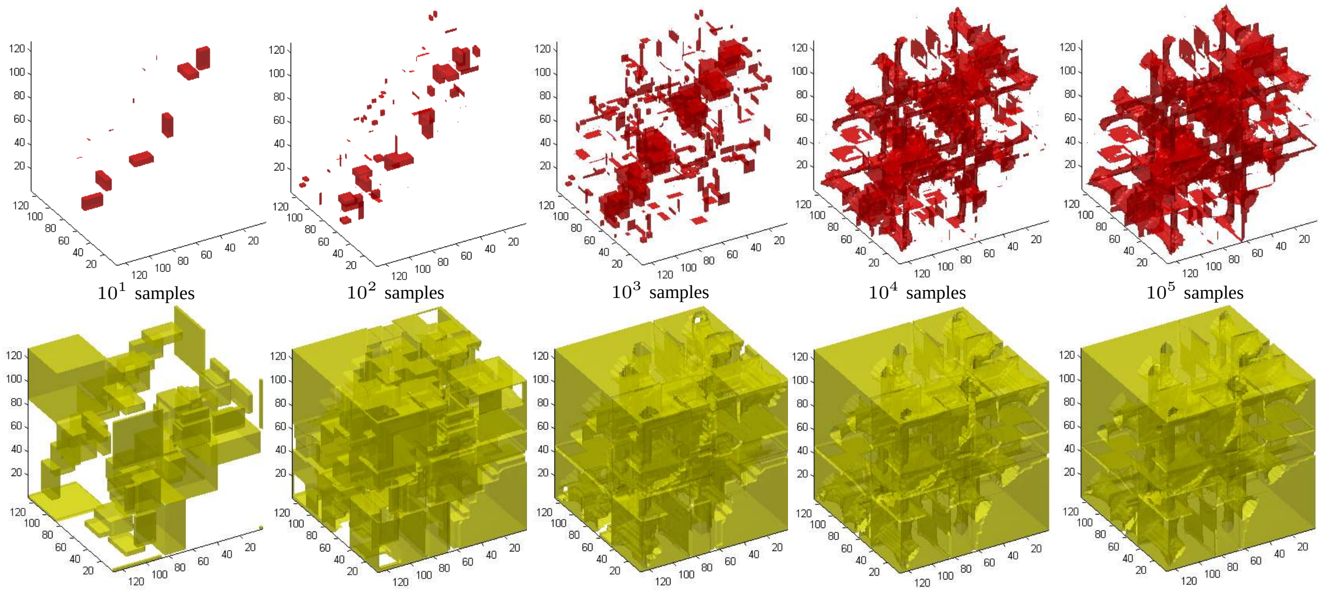

Fig. 9. Evolution in the generation of the grasp space for Example 2. Up: FC grasp space. Down: non-FC grasp space.

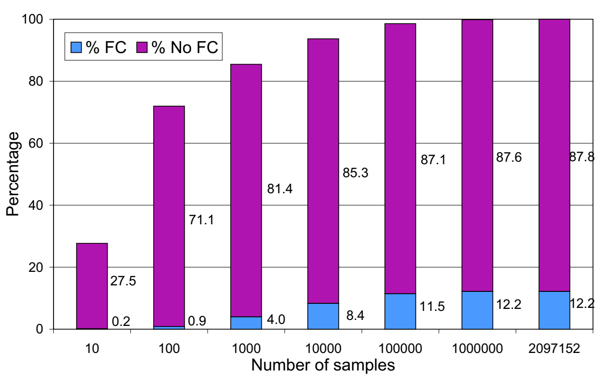

Fig. 10. Evolution of the grasp space generation for Example 2.

NGR involves 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 algorithms presented in the paper have been implemented and same application examples are given. The procedures are fully valid for 3D objects with high-dimensional grasp spaces, however, the application to 3D objects requires an efficient way to save the data (the grasp space has a high dimensionality, for instance it is 8-dimensional for a 4-finger frictional grasp on a 3D object). The effect of different sampling methods will be addressed as future work (e.g. a classical grid search [15] or a deterministic sampling method [16]).

The generation of the grasp space with the proposed approach is useful for different manipulation applications, one of the most relevant is regrasping an object for manipulation purpose, moving the fingers along the object surfaces while keeping the force-closure property. This particular application may not require the total exploration of the grasp space. Work in this line is currently under development.

REFERENCES

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

[2] Y. Liu, "Computing n-finger form-closure grasps on polygonal objects," *Int. J. Robotics Research*, vol. 19, no. 2, pp. 149–158, 2000.

[3] N. Niparnan and A. Sudsang, "Computing all force-closure grasps of 2D objects from contact point set," in *Proc. IEEE/RSJ IROS*, 2006, pp. 1599–1604.

[4] T. Omata, "Finger position computation for 3-dimensional equilibrium grasps," in *Proc. IEEE ICRA*, 1993, pp. 216–222.

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

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

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

[9] 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.

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

[11] 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.

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

[13] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Ratón, Florida: CRC Press, 1994.

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

[15] S. Lavalle, M. Branicky, and S. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *Int. J. Robotics Research*, vol. 23, no. 7-8, pp. 673–692, 2004.

[16] J. Rosell, M. Roa, A. Perez, and F. Garcia, "A general deterministic sequence for sampling d-dimensional configuration spaces," *J. Intelligent and Robotic Systems*, vol. 50, no. 4, pp. 361–374, 2007.