UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Exploration of the grasp space using independent contact and non-graspable regions

Máximo Roa, Raúl Suárez, Jan Rosell.

IOC- Divisió de Robòtica

IOC-DT-P-2008-05
Abril 2008

Institut d'Organització i Control
de Sistemes Industrials

Exploration of the Grasp Space Using Independent Contact and Non-Graspable Regions

Máximo Roa*, Raúl Suárez†, Jan Rosell

Institute of Industrial and Control Engineering (IOC)

Technical University of Catalonia (UPC), Barcelona, Spain

E-mail: maximo.roa@upc.edu, raul.suarez@upc.edu, jan.rosell@upc.edu

April 2008

Abstract

This report 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.

Keywords: Grasp space, independent contact regions, non-graspable regions.

1 Introduction

Grasp planning searches for a desirable location of the fingers on the surface of an object, for instance, to assure the equilibrium of the object, or to fully restrain the object to resist the influence of external disturbances. To assure the immobility of 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 to synthesize precision grasps (i.e. grasps formed by a set of finger contact points on the object surface) for 2D [2] [3] and 3D objects [4] [5].

To provide robustness in front of finger positioning errors for an object grasp, the computation of independent contact regions (ICRs) on the object boundary was introduced [6]. Each finger can be positioned in each ICR assuring in this way a force-closure (FC) grasp,

*Department of Mechanical and Mechatronic Engineering - 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).

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

independently of the exact position of each finger. The computation of ICRs has been solved for 2D [7] and 3D objects [4] [8]. The ICRs have also been used to determine contact regions on 3D objects based on initial examples, although the results depend on the chosen example [9]. To generate a procedure applicable to objects with an arbitrary shape, the computation of ICRs has been tackled for 2D [10] and 3D [11] discrete objects, i.e. objects described with a mesh of surface points, and with frictional and frictionless contacts. Based on the ICRs, this report introduces the concept of non-graspable regions (NGRs), defined such that every finger can be positioned inside its corresponding NGR and a non-FC grasp will always result with independence of the exact position of each finger.

Most of the works above-mentioned focus on the synthesis of one grasp configuration that optimizes a particular criterion. However, in applications such as manipulation and regrasp planning it is useful to know all the possible FC grasp configurations (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-fingers 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 report presents a method to generate the grasp space for discrete objects using NGRs and ICRs, i.e. all the FC and non-FC grasps are computed for the object.

The rest of the report is organized as follows. Section 2 provides the required background on FC grasps and grasp spaces. Section 3 describes the approach to generate the grasp space, and Section 4 presents the algorithms to compute the independent contact regions and non-graspable regions starting with a sample FC or non-FC grasp, respectively. Section 5 illustrates the implementation of the approach on two different 2D objects. Finally, Section 6 summarizes the work and discusses some future applications.

2 Framework

2.1 Assumptions

This report addresses the problem of generating the grasp space for a discrete object using the concepts of independent contact regions and non-graspable regions. The work is based on three assumptions. First, the contact between the object and the fingers is punctual, and the friction between them is characterized by Coulomb's law, with a friction coefficient μ ($\mu = 0$ accounts for the frictionless case). Second, the object surface is discretized in a sufficiently large set Ω of points $\boldsymbol{p}_i$, described by one or two parameters u that provide the position of the point on the surface of a 2D or 3D object, respectively; moreover, each point has an associated unitary normal direction $\hat{\boldsymbol{n}}_i$ pointing toward the interior of the object. Third, each point is connected with a set of neighboring points forming a mesh; the number of neighbors may be arbitrarily selected, and has no influence on the proposed approach (i.e. different types of meshes are valid).

2.2 Grasp space and force-closure conditions

An n-finger grasp G is defined as the set of parameters u_i that determine the position of the fingers on the surface of the grasped object, i.e. $G = \{u_1, \ldots, u_p\}$, with $p = n$ for 2D objects or $p = 2n$ for 3D objects. The p-dimensional space defined by the p parameters that represent the position of the possible contact points is called the grasp space (also known as grasp configuration space or contact space) [8].

A unitary force $\boldsymbol{f}_i$ applied on the object at the point $\boldsymbol{p}_i$ generates a torque $\boldsymbol{\tau}_i = \boldsymbol{p}_i \times \boldsymbol{f}_i$; the force and the torque are grouped together in a wrench vector given by $\boldsymbol{\omega}_i = (\boldsymbol{f}_i, \boldsymbol{\tau}_i)^T$. The wrenches produced by the forces applied at the contact points on the object are grouped in a wrench set W. For frictionless grasps, $W = \{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_n\}$, as the grasp forces can only be applied in the direction normal to the object. For frictional grasps, the grasp forces lie inside a friction cone that can be linearized with an m-side polyhedral convex cone. The grasping force at the contact point is given by

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

with $\boldsymbol{s}_{ij}$ representing the normalized vector of the j-th edge of the convex cone. The wrench produced by the force $\boldsymbol{f}_i$ 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 $\boldsymbol{\omega}_{ij}$ are called the primitive contact wrenches. Therefore, $W = \{\boldsymbol{\omega}_{11}, \ldots, \boldsymbol{\omega}_{1m}, \ldots, \boldsymbol{\omega}_{n1}, \ldots, \boldsymbol{\omega}_{nm}\}$ for frictional grasps.

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 the wrench set, $CH(W)$ [12]. The FC condition is applied in this work using the following lemma [11].

Lemma 1: Let G be a grasp with a set W of contact wrenches, $\mathcal{I}$ be the set of strictly interior points of $CH(W)$, and H be a supporting hyperplane of $CH(W)$ (i.e. a hyperplane containing one of the facets of $CH(W)$). The origin O of the wrench space satisfies $O \in \mathcal{I}$ if and only if any point $P \in \mathcal{I}$ and O lie in the same half-space for every H of $CH(W)$.

From *Lemma 1*, checking whether a given point $P \in \mathcal{I}$ and the origin O lie in the same half-space defined by each supporting hyperplane H is enough to prove whether O lies inside $CH(W)$, i.e. to prove the FC property for the grasp G. P is chosen as the centroid of the primitive contact wrenches, which is always an interior point of $CH(W)$; therefore, the FC test used in this report checks whether the centroid P and the origin O lie on the same side for all the supporting hyperplanes of $CH(W)$.

3 Generation of the grasp space

The generation of the grasp space is based on the concepts of Independent Contact Regions (ICRs) and of Non-Graspable Regions (NGRs). The ICRs and NGRs are regions such that each finger can be positioned anywhere inside its corresponding region and 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 FC grasps (or non-FC grasps) on the object. This action can be repeated until a useful portion of the grasp space is already labeled (for instance for grasp or regrasp planning purposes) or simply until the whole grasp space is labeled. The algorithm is as follows:

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.

   Endif
3. If the grasp space is not fully labeled yet, go to Step 1. Otherwise, the algorithm returns the grasp space.

The sampling method used to generate samples for 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.

4 Computation of the independent contact and non-graspable regions

4.1 Independent contact regions

This subsection summarizes the procedure previously presented in [11] 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 S_i, the search zone containing the ICR for the grasp point $\boldsymbol{p}_i$; S_i is the intersection of the open half-spaces H''^{+}_k that contain the point $\boldsymbol{p}_i$. The ICR is determined by the set of neighbor points of $\boldsymbol{p}_i$ such that at least one of its primitive wrenches falls into the corresponding search zone S_i. The steps in the algorithm are:

Algorithm 2: Search of the independent contact regions

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

For each contact point $\boldsymbol{p}_i$, $i = 1, \ldots, n$, do

1. Build the hyperplanes H''_k parallel to every F_k, and containing the origin O.
2. Let $S_i = \bigcap_k H''^{+}_k$ with H''^{+}_k the open half-spaces such that $\boldsymbol{p}_i \in H''^{+}_k$ (i.e. $\boldsymbol{\omega}_i \vee \boldsymbol{\omega}_{i1} \vee \ldots \vee \boldsymbol{\omega}_{im} \in S_i$).
3. Initialize $I_i = \{\boldsymbol{p}_i\}$. Label the points in I_i as open.
4. While there are open points $\boldsymbol{p}_j \in I_i$

   For every neighbor point $\boldsymbol{p}_s$ of $\boldsymbol{p}_j$

   If $\boldsymbol{\omega}_s \vee \boldsymbol{\omega}_{s1} \vee \ldots \vee \boldsymbol{\omega}_{sm} \in S_i$

   $I_i = I_i \cup \{\boldsymbol{p}_s\}$

   Label $\boldsymbol{p}_s$ as open.

   Endif

   Endfor

   Label $\boldsymbol{p}_j$ as closed.

   Endwhile
5. Return the set of points I_i (i.e. the ICR 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 the point $\boldsymbol{p}_i$, and forms the search zones S_i; note that because of the geometrical construction the selection of any arbitrary point from each S_i always generates a FC grasp. Step 4 is the most complex step in the algorithm; every checked point requires its classification with respect to the number of hyperplanes H_k that contain at least one primitive wrench for the contact point $\boldsymbol{p}_i$.

The number of points in each I_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 vectors around the contact location. Finally, considering the ICRs for each finger, several grasps can be formed when each finger is placed in a different position inside its ICR; the geometrical procedure assures that all these grasps satisfy $O \in CH(W)$.

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 b); 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

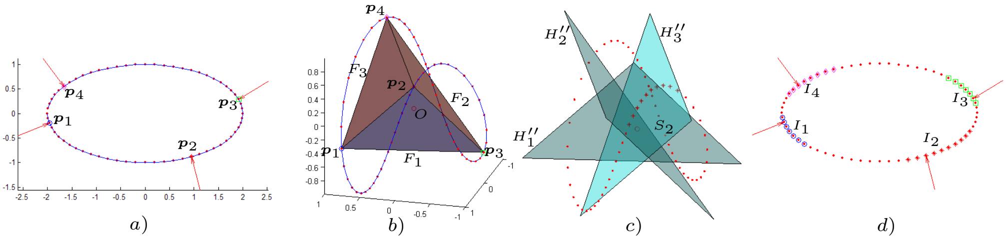


Figure 1. Search of the independent contact regions for a discretized ellipse: a) Initial FC grasp, b) FC grasp in the wrench space, c) Search of the ICR for the point $\boldsymbol{p}_2$, d) ICRs on the ellipse.

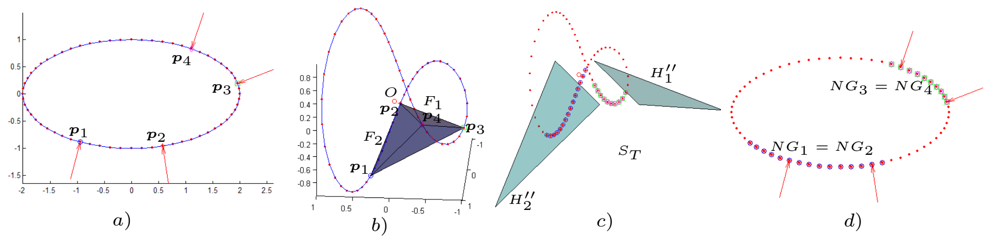


Figure 2. Search of the non-graspable regions for a discretized ellipse: a) Initial non-FC grasp, b) Non-FC grasp in the wrench space, c) Search of the NGRs for the grasp, d) NGRs on the ellipse.

the wrenches corresponding to the neighbor points of $\boldsymbol{p}_2$ that fall in the search zone 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 positions for each finger inside its corresponding ICR.

4.2 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. The approach determines the subset T of hyperplanes H_k'' that leave all of $CH(W)$ in the same open half-space (i.e. if a H_k'' intersects $CH(W)$ then the hyperplane 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 $H_k''^{+}$. The NGR is determined by the set of neighbor points of $\boldsymbol{p}_i$ such that all of its primitive wrenches lie in the search zone S_T. The steps in the algorithm are:

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 every F_k, and containing the origin O.
2. Let $S_T = \bigcap H_k''^{+}$ with $H_k''^{+}$ the open half-spaces such that $CH(W) \subset H_k''^{+}$ (i.e. $\boldsymbol{\omega}_i \wedge \boldsymbol{\omega}_{i1} \wedge \ldots \wedge \boldsymbol{\omega}_{im} \in H_k''^{+}$ for every $\boldsymbol{p}_i$)
3. Initialize $NG_i = \{\boldsymbol{p}_i\}$. Label the points in NG_i as open.

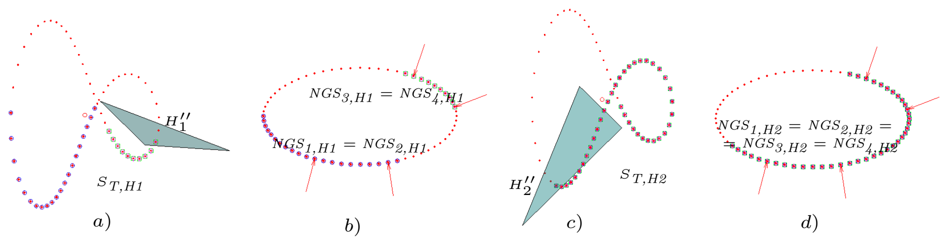


Figure 3. Search of the non-graspable sets for the previous example: a) Hyperplane H_1'' and NGSs in the wrench space, b) Sets NGS_{H1} on the ellipse, c) Hyperplane H_2'' and NGSs in the wrench space, d) Sets NGS_{H2} on the ellipse.

4. For every contact point $\boldsymbol{p}_i$

    While there are open points $\boldsymbol{p}_j \in NG_i$

    For every neighbor point $\boldsymbol{p}_s$ of $\boldsymbol{p}_j$

    If $\boldsymbol{\omega}_s \wedge \boldsymbol{\omega}_{s1} \wedge \ldots \wedge \boldsymbol{\omega}_{sm} \in S_T$

    $NG_i = NG_i \cup \{\boldsymbol{p}_s\}$

    Label $\boldsymbol{p}_s$ as open.

    Endif

    Endfor

    Label $\boldsymbol{p}_j$ as closed

    Endwhile

5. Return the sets of points NG_i (i.e. the NGRs for each contact point $\boldsymbol{p}_i$).

Again, the algorithm is very simple. Note that as $O \notin S_T$, choosing every possible combination of one point from each NG_i always generates a $CH(W)$ that 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 b). 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 neighbor points of every $\boldsymbol{p}_i$ that fall in the search zone S_T are depicted. Fig 2d show the NGRs for the 4 grasp points; note that the NGRs coincide for $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, and for $\boldsymbol{p}_3$ and $\boldsymbol{p}_4$; 22500 different non-FC grasps may be obtained from the possible combinations of positions for every finger inside its corresponding NGR.

To maximize the number of non-FC grasps identified in each call to Algorithm 3, note that every hyperplane H_k'' that fulfills the condition on Step 2 (i.e. $CH(W) \subset H_k''^{+}$) may generate its own set of NGRs, called NGSs. For each H_k'' in T, the search region is redefined as $S_T = H_k''^{+}$, and the corresponding NGSs are computed using Steps 3 to 4 in Algorithm 3. For instance, in the previous example (Fig. 2), two hyperplanes H_k'' are considered to compute the NGRs. Fig. 3a and c show the two hyperplanes and the corresponding NGSs in the wrench space, and Fig. 3b and d show the NGSs on the ellipse. The hyperplanes H_1'' and H_2'' define the non-graspable regions NGS_{H1} and NGS_{H2} that provide 44100 and 2313441 different non-FC grasps, respectively. The equivalence of the NGRs in Fig. 2d with the NGSs in Fig. 3 is given by $NG_i = NGS_{i,H1} \cap NGS_{i,H2}$.

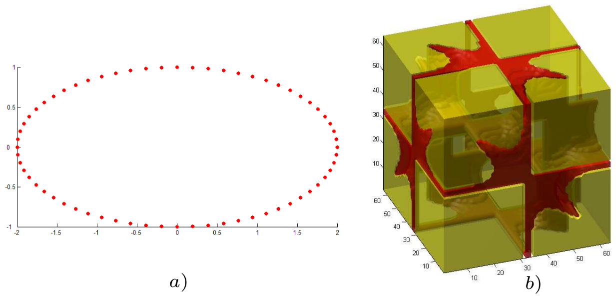


Figure 4. Example 1: a) Ellipse, b) Grasp space.

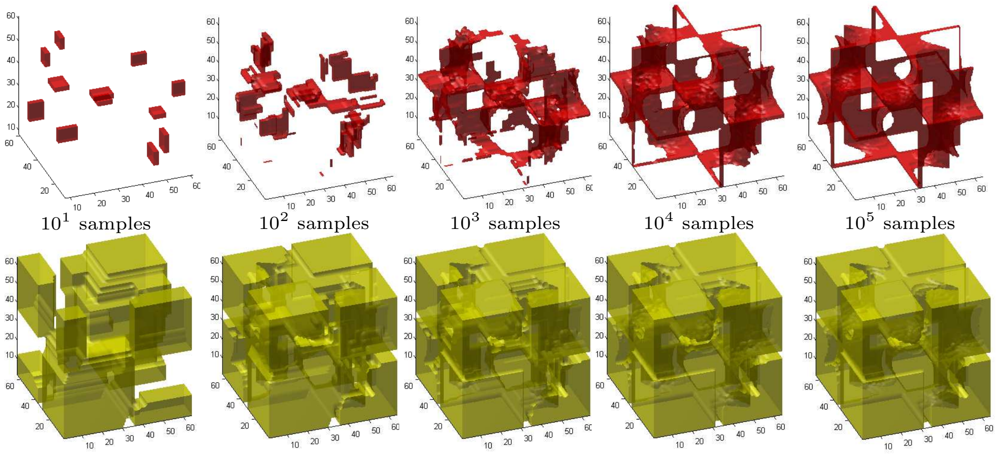


Figure 5. Evolution in the generation of the grasp space for the first example. Up: FC grasp space. Down: non-FC grasp space.

5 Examples

To illustrate the application of the proposed approach, the algorithms were implemented in Matlab on a Pentium IV 3.2 GHz computer. The following examples show the generation of the grasp space for 3-finger frictional grasps on two 2D objects. This examples were selected for ease of visualization, as the corresponding grasp space is three-dimensional.

5.1 Example 1

The first example uses an ellipse discretized with 64 points along its boundary, as shown in Fig. 4a; the grasp space contains $64^3 = 262144$ 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. The evolution in the generation of the grasp space using Algorithm 1 is presented in Fig. 5. The grasp space has some symmetries, as any grasp $G = \{u_1, \ldots, u_p\}$ accounts for 6 different grasps (the total number of possible permutations of the fingers on the ellipse while keeping the same contact points); therefore, an ICR or NGR region 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

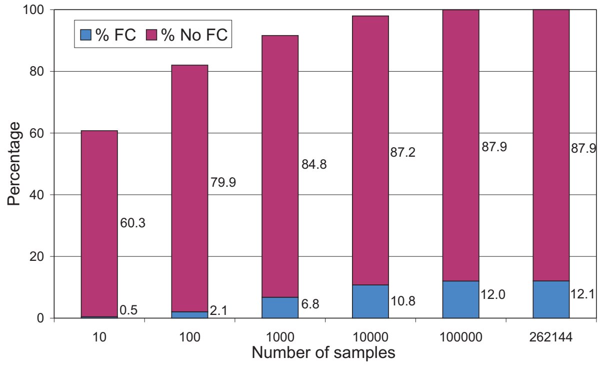


Figure 6. Percentage evolution of the grasp space generation for the first example.

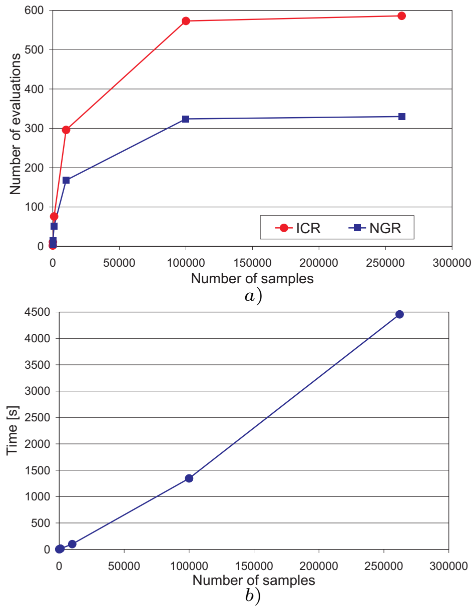
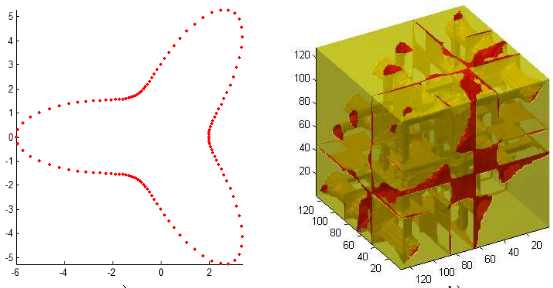


Figure 7. Parameters in the grasp space generation: a) Number of evaluations of ICRs and NGRs, b) Sampling time.

whole space has been already explored with just 100 samples; with 10^4 samples (3.8% of the total number of grasps), 98% of the grasp space has been generated. Fig. 7a presents the number of calls to Algorithms 1 and 2, i.e. the number of evaluations of ICR and NGR regions; Fig. 7b presents the time required for the generation of the space.

5.2 Example 2

In the second example, the object is defined by a closed parametric curve presented in [13], discretized with 128 points on its boundary, as shown in Fig. 8a. Fig. 8b shows the total grasp space for this figure; it contains $128^3 = 2097152$ grasps, with 12.2% and 87.8% of the space corresponding to 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 (the results correspond to the average of 20 trials). The behavior of the algorithm to generate the grasp space is the same



a) *b*)

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

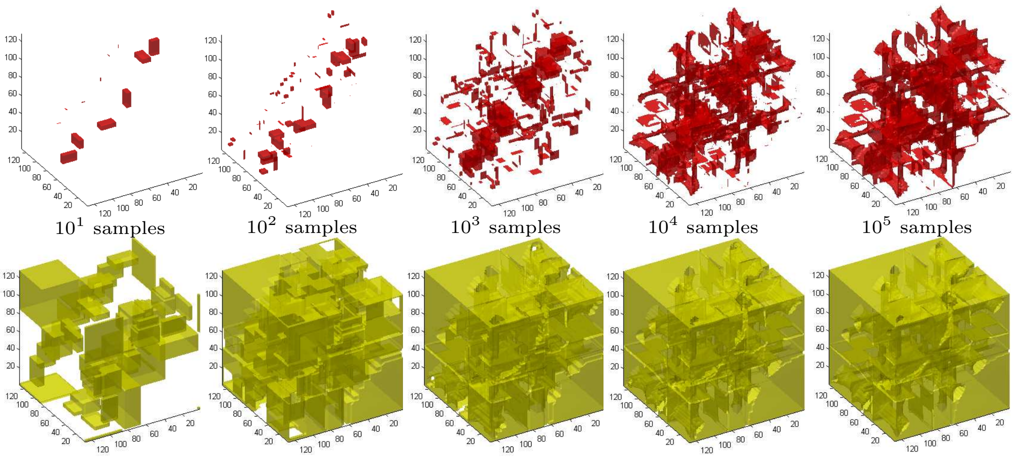


Figure 9. Evolution in the generation of the grasp space for the second example. Up: FC grasp space. Down: non-FC grasp space.

as for the previous example: for a low number of samples, a large portion of the grasp space is covered, e.g. for 10000 samples (0.48% of the total grasp space) 93.7% of the space has been generated, in $\sim$ 1000 seconds.

6 Conclusions

This report has presented a method to generate the grasp space for 2D and 3D discrete objects, valid 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. A more efficient exploration method is proposed, based on the concepts of independent contact regions (ICRs) and non-graspable regions (NGRs). The ICRs have been used previously in several works, but the concept of NGRs is new and introduced in this report; the NGRs are defined as regions on the object boundary such that when each finger is positioned inside its corresponding NGR, a non-FC grasp is always obtained, with independence of the exact position of each finger.

The proposed approach uses a sampling method to obtain a grasp sample. If the sample is a FC grasp, the ICRs are computed; if it is a non-FC grasp, the NGRs are computed. Every ICR or NGR detects 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

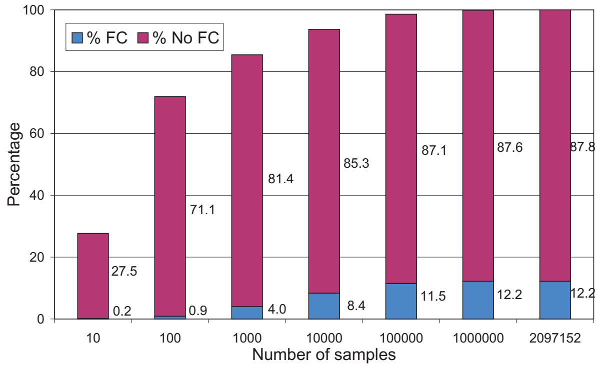


Figure 10. Percentage evolution of the grasp space generation for the second example.

in the report have been implemented, and for ease of visualization they are illustrated with the 3-finger frictional grasp of two 2D objects, which generates a three-dimensional grasp space. The procedures are fully valid for 3D objects with high-dimensional grasp spaces; however, the implementation of the method for 3D objects requires an efficient way to save the data, as the grasp space has a high dimensionality (e.g. the grasp space is 8-dimensional for a 4-finger frictional grasp on a 3D object). The effect of different sampling methods (e.g. a classical grid search [14] or a deterministic sampling method [15]) will be addressed in the future.

The generation of the grasp space has several applications in manipulation of objects, as the method provides in a short time a large number of FC and non-FC grasps. Moreover, the approach may be used in the regrasp of an object, i.e. to move the fingers on the object to change from one FC grasp to another one; this particular application does not require the total exploration of the grasp space. These works are currently under development.

References

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

[2] Y.H. Liu. Computing n-finger form-closure grasps on polygonal objects. *Int. J. Robotics Research*, 19(2):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*, pages 1599–1604, 2006.

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

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

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

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

[8] J. Ponce and B. Faverjon. On computing three-finger force-closure grasps of polygonal objects. *IEEE Trans. Robotics and Automation*, 11(6):868–881, 1995.

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

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

[11] M. Roa and R. Suarez. Independent contact regions for frictional grasps on 3D objects. In *Proc. IEEE ICRA 2008*, 2008.

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

[13] Y.B. Jia. Computation on parametric curves with an application in grasping. *Int. J. Robotics Research*, 23(7-8):825–855, 2004.

[14] S.M. Lavalle, M.S. Branicky, and S.R. Lindemann. On the relationship between classical grid search and probabilistic roadmaps. *Int. J. Robotics Research*, 23(7-8):673–692, 2004.

[15] J. Rosell, M. Roa, A. Perez, and F. Garcia. A general deterministic sequence for sampling d-dimensional configuration spaces. *J. Intelligent and Robotic Systems*, 50(4):361–374, 2007.