

# Grasp Planning with Soft Hands using Bounding Box Object Decomposition

M. Bonilla<sup>1</sup>, D. Resasco<sup>1</sup>, M. Gabiccini<sup>1,2,3</sup> and A. Bicchi<sup>1,2</sup>

**Abstract**—In this paper, we present a method to plan grasps for soft hands. Considering that soft hands can easily conform to the shape of the object, with preference to certain types of basic geometries and dimensions, we decompose the object into one type of these geometries, particularly into Minimal Volume Bounding Boxes (MVBBs), which are proved to be efficiently graspable by the hand we use. A set of hand poses are then generated using geometric information extracted from such MVBBs. All hand postures are used in a dynamic simulator of the PISA/IIT Soft Hand and put on a test to evaluate if a proposed hand posture leads to a successful grasp. We show, through a set of numerical simulations, that the probability of success of the hand poses generated with the proposed algorithm is very good and represents an evident improvement with respect to our previous results published in [1].

## I. INTRODUCTION

The increase of robot capabilities to actively execute tasks and modify surrounding scenarios, thereby reaching versatile goals, is tightly linked to the ability to generate stable grasps for objects that are even unknown to the robot. A key tool to this ability is the faculty of selecting a successful grasp without the need of identifying *a priori* an object, but only based on its composition of shape primitives with known associated grasps, i.e. resorting to *part-based grasps*.

To this sake, assuming that distorted and/or scattered clouds of 3D points of an object are given — here we assume the segmentation algorithm robust enough to distinguish the manipulandum from the environment — two basic ingredients are needed: (i) a suitable algorithm enabling the representation of these points as a set of shape primitives (e.g., boxes, spheres or cylinders), and (ii) a strategy that takes advantage of the simple primitive shape representation of parts to associate grasps to them.

This approach is well entrenched in the grasp planning community, as testified by [2], [3] and [4]. A not nearly complete classification of the vast literature on this topic can be attempted on the basis of the types of primitive shapes employed for object decomposition. Many bottom-up approaches — starting from point clouds they synthesize object part shapes — use superquadrics (SQs). SQs are parametrizable models that offer a large variety of different shapes. However, to approximate object parts, only superellipsoids out of the groups of SQs are employed in practice, as only these represent closed shapes. In these group we can

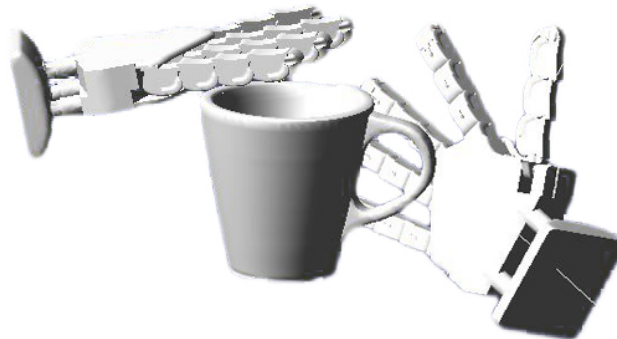


Fig. 1. The main idea of this paper is to find different hand poses with high probability of being successful when grasping objects

recall [5], [6] and [7], not mention only a few. However, the more complex the shape is, there higher the number of SQs that have to be instantiated to accurately represent the different parts, and the immense parametrization capabilities lead to highly inefficient algorithms. To overcome these difficulties, beside the *region-growing* (bottom-up) strategy which proved to be not very effective in practice [7], the *split-and-merge* (split: top-down, merge: bottom-up) technique was proposed. Even if this technique allowed to better cope with irregular and unorganized data, the high sensitivity to noise and outliers of SQs approximations, make their wide use still impractical.

Following a mid-level solution, according to a purely top-down strategy, a *fit-and-split* algorithm based on Minimum Volume Bounding Box (MVBB) algorithm to object decomposition for grasp planning was originally proposed in [8]. The method consists on iteratively build MVBB of points resulting from splitting the point cloud of the object. The split procedure is performed in such a way that the sum of the two areas resulting from the convex region of each set of points is minimized. An algorithm for grasp planning for fully actuated hands using MVBB decomposition of object was proposed in [9].

In this paper, we adopt a modified version of the MVBB algorithm for object decomposition in [8], as it presents the following properties: (i) it is very efficient and can accommodate scattered 3D points delivered by arbitrary 3D sensors; (ii) the outcome constellation of boxes is quite insensitive to noise. However, with respect to [9] and [10], where 2D grasp hypotheses are made and evaluated on point projections onto box faces and assume, from the outset, the use of rigid and fully-actuated robotic hands or grippers, in

<sup>1</sup>Research Center "E. Piaggio", Università di Pisa, Largo Lucio Lazzarino 1, 56122 Pisa, Italy

<sup>2</sup>Department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy

<sup>3</sup>Dipartimento di Ingegneria Civile ed Industriale, Largo Lucio Lazzarino 1, 56122 Pisa, Italy

the present work we propose a shift of perspective in the creation of grasp hypotheses as we employ a compliant and adaptive hand, the Pisa/IIT SoftHand.

New developments in robotic hands [11], [12] have shown that the correct combination of underactuation and embedded compliance is key to have robotic hands with dexterity comparable to the human hand. In practice, thanks to their mechanism, these hands adapt to the shape of the object while executing a grasp. This behavior brings about a shift of paradigm in grasp planning [1], since old algorithms used to find points or regions in the object to establish a contact can not be directly applied. Considering the mentioned behavior, a rough approximation of the geometry of the object via decomposition into bounding boxes *is enough* to generate candidate hand poses to grasp an object. The possibility to rely on the adaptive behavior of the hand embedded in its the mechanical design, alleviates the task of planning the exact finger placement on the object, and can easily absorb shape primitives approximations due to noisy data, as well as limitations in the representation itself. The framework allows for an easier handling of approximation inaccuracies through haptic feedback and grasp controllers for online corrections.

With a reduced burden to plan detailed finger placements, the box set representation of an object, also ease the mapping of complex actions to box and/or box distribution properties. For example, as also mentioned in [10], in order to *pick-up* an object and place it somewhere, it may intuitively be a good option to grasp the *largest box*. Instead, in order to show the same object to a camera to gather more views, it may be preferable to grasp the object from the *outermost box*, and or, from a box that allows a pincher grasp, so to minimize occlusions caused by hand parts.

In this paper, we recall the method to decompose objects into MVBB presented in [8]. Then, considering the adaptability of soft hands, we present a strategy to propose grasp postures to grasp a set of kitchenware objects [13] previously decomposed into MVBBs. The performances of the method are compared with those presented in [1] and, through simulations performed with the MBS software ADAMS<sup>TM</sup> [14], we show that with the strategy presented in this paper we highly increase the successful rate of grasps for the same objects.

### A. Organization

Section II recalls the algorithm used to decompose the object into MVBB. In Section III, the method developed in this paper to generate hand postures to grasp an object is presented. In Section IV, we present the analysis of the simulations performed with a set of kitchenware objects and, finally, Section V presents the conclusions.

## II. BOUNDING BOX DECOMPOSITION

Algorithm 1 was presented in [9]. The idea is to decompose the object in MVBBs minimizing the volume of the boxes which fit partial point clouds. The algorithm takes a point cloud of an object ( $points^{3D}$ ) and approximates it with a MVBB. The points are then projected onto three

planes which are the non-opposite faces of the box. Then, using Algorithm 2, the points are split in two sets. The split is performed for each of the projected points ( $faces$ ), and for each of the two projection axes. After that, the points are approximated with a box and its area is computed. At the end, the algorithm returns the point and the direction minimizing the mentioned area. The split is then performed for the set of all 3D points: it results in two boxes with a set of 3D points. The reduction rate of the volume of the two new boxes compared with the original is then compared with a user-given parameter  $t$  to judge the split useful or not. In case it is found useful, the split is performed and each of the boxes are consider as new point clouds to repeat the procedure on; the algorithm is stopped otherwise.

---

### Algorithm 1 Approximate the object in MVBB

---

```

1: procedure BOXAPPROXIMATE( $faces, points^{3D}$ )
2:    $box \leftarrow FindBoundingBox(points^{3D})$ 
3:    $faces \leftarrow nonOppositeFaces(box)$ 
4:    $(p, q) \leftarrow split(FindBestSplit(faces, points^{3D}))$ 
5:   if ( $percentualVolume(p + q, box) < t$ ) then  $\triangleright t$  is
      a stop criteria
6:      $BoxApproximate(p)$ 
7:      $BoxApproximate(q)$ 
8:   end if
9: end procedure

```

---



---

### Algorithm 2 Split the point cloud

---

```

1: procedure FINDBESTSPLIT( $faces, points^{3D}$ )
2:   for  $i \leftarrow 1$  to 3 do
3:      $p^{2D} \leftarrow projects(points^{3D}, faces[i])$ 
4:     for  $x \leftarrow 1$  to  $width(faces[i])$  do
5:        $(p1, p2) \leftarrow verticalSplit(p^{2D}, x)$ 
6:        $a1 \leftarrow boundArea^{2D}(p1)$ 
7:        $a2 \leftarrow boundArea^{2D}(p2)$ 
8:       if ( $a1 + a2 < minArea$ ) then
9:          $minArea \leftarrow (a1 + a2)$ 
10:         $bestSplit \leftarrow (i, x)$ 
11:      end if
12:    end for
13:    for  $x \leftarrow 1$  to  $height(faces[i])$  do
14:       $(p1, p2) \leftarrow verticalSplit(p^{2D}, y)$ 
15:       $a1 \leftarrow boundArea^{2D}(p1)$ 
16:       $a2 \leftarrow boundArea^{2D}(p2)$ 
17:      if ( $a1 + a2 < minArea$ ) then
18:         $minArea \leftarrow (a1 + a2)$ 
19:         $bestSplit \leftarrow (i, y)$ 
20:      end if
21:    end for
22:  end for
23: end procedure

```

---

Fig. 2 shows a comparison of different values of  $t$ . Depending on the task that we want to perform, this parameter can assume different values. For example, if we want to

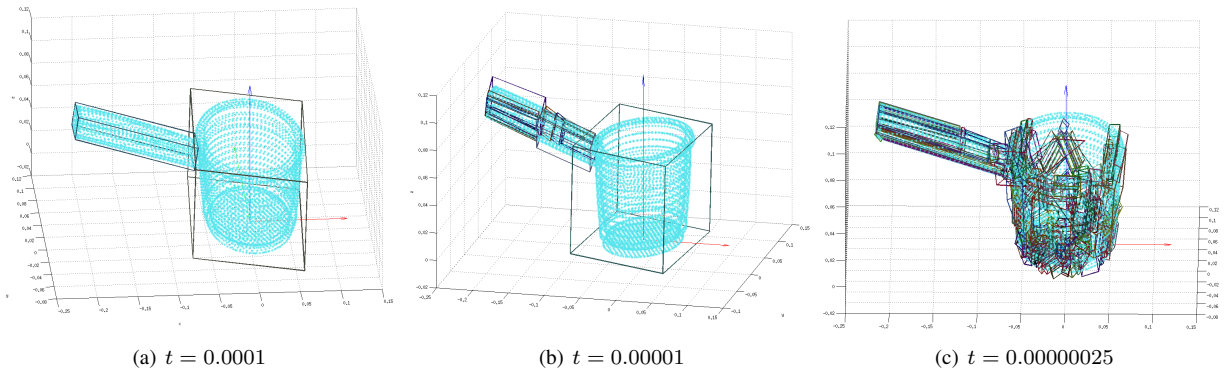


Fig. 2. Comparison of the MVBB generated by the algorithm 1 using different values of  $t$ .

grasp objects from handles, like in cups or pots, they can be isolated using a small value of  $t$ , see Fig. 2(a). Similar values can be used to grasp a cup from above for example. On the other hand, if we want to explore more deeply the geometry of the object to, for example, being able to grasp edges in pinch grasp configurations, the parameter  $t$  must be increased, see Fig. 2(b) and Fig. 2(c).

In practice, the selection of the parameter  $t$  is related to the translation of high level task specifications to low level grasp actions. Let us consider an example in which a robot has to pick up a pot to pour the content into another glass. In this case, a convenient choice is to grasp the pot from the handle, as shown in Fig 3(a). Therefore, in this case a very fine object decomposition is not necessary. On the other hand, if we consider the task passing an object from one hand to another in a bi-manual manipulation setting, the selected box to be grasped could be one in the border of the cooker body, see Fig. 3(b), such that the second arm has more options to decide where to grasp the object without collisions with the first hand. Thus, in this case, a finer object decomposition is beneficial.

### III. PROPOSING GRASP POSES

The aim of this section is to explain how we align the hand with respect to the object in order to grasp it. We consider the orientation of each of the MVBBs and the orientation of the object itself. The orientation of the boxes come from the principal axis defined by the Principal Component Analysis (PCA) performed inside the *FindBoundingBox* function. The inclusion of the PCA is one of the differences with

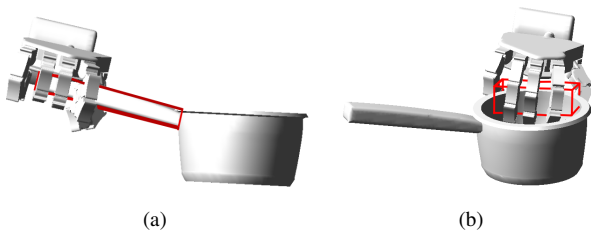


Fig. 3. The selection of the box to grasp depend on the high level task specification.

respect to the original algorithm in [9], and makes the algorithm invariant to the reference frame of the point cloud. In the other hand, the frame attached to the point cloud of the object is defined always with the  $z$  axis in the normal direction of its base. Then, the  $x$  axis is oriented to some features of interest (a handle, for example) and parallel to the plane of the object base. The origin is placed always in the intersection of the middle axis of the object and the base plane. Once the object is decomposed into MVBBs, the next step is to select a box to grasp. There are many criteria to do this, the most promising and useful depending on the task that the robot has to perform once the object is grasped. In this paper, we start generating hand poses from the outermost box. This choice is driven by our first priority of just grasping the object in a successful manner — most probably in a power grasp configuration, as the hand is just closed to a certain extend — for, e.g., clearing a table. Once a MVBB is selected, the procedure followed to find the transformation  $T_O^H$  describing the pose of the hand with respect to the object is the following:

- 1) Align the  $x$  axis of the hand parallel the longest side of the MVBB.
- 2) Align the  $z$  axis of the hand with the axis of the box which has the smallest angle with respect to the  $z$  axis

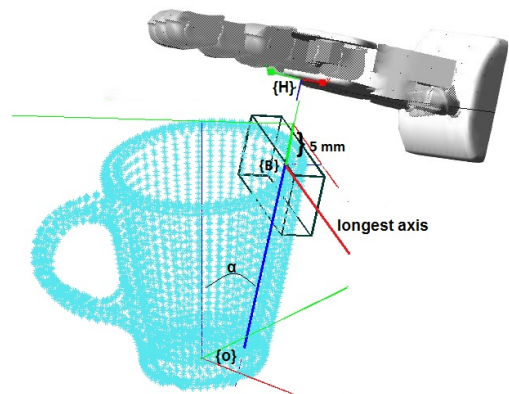


Fig. 4. Graphical explanation of the procedure performed to align the hand with each bounding box.

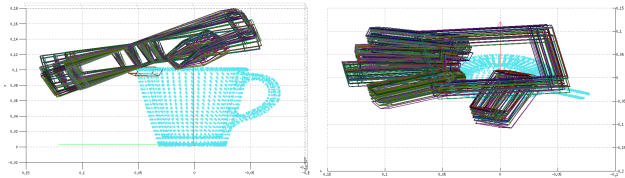


Fig. 5. One example of the free collision hand poses generated for one of the box of the cup.

of the hand.

- 3) Compute the orientation of the  $y$  axis to form a right-handed frame.

From this procedure, we can generate the rotation matrix  $R_O^H$  defining the orientation of the hand frame  $H$  with respect to the object frame  $O$ . The frame  $H$  is placed 5 mm out of the MVBB, in the negative direction of the  $z$  axis defined previously. The procedure is explained graphically in Fig. 4.

From the previous steps we can see that, from steps 1) and 2), the selected axis can be the same. In this case, we still align the  $x$  axis with the longest side of the MVBB. Instead, the  $z$  axis is aligned with the axis of the box which has the smallest angle with respect to the vector connecting the centroid of the MVBB with the object centroid.

#### A. Pose Variations

The previous procedure generates just a single hand configuration. However, once a MVBB is generated, there is a large number of possibilities to grasp it.

In order to generate more variations for a box, we first set the range of motion in which we can move the hand, translating a distance  $x_t$  and rotating by an angle  $\alpha_t$ , both along the longest axis of the box, while still not colliding with the object. Fig. 6 shows the random variations created for the cup. Variables  $x_t$  and  $\alpha_t$  generates a 2D space, with high probability of being free collision, from where we pick a random point, with uniform distribution, and the check for collision. If this configuration is collision free, then it is a candidate pose to grasp the object. In this work, we generate 40 random configurations for each box and considered the first 5 boxes on the object, thus for each object there are 200 candidate poses.

This procedure constitutes one of the differences with respect to the method proposed in [1], where the authors did not consider collisions in the procedure previous to the simulations. As they explained in the paper, they had a high percentage of failures just because from the very beginning of the simulations, there were many collisions with the object.

In this work, we used a simple collision detector. The hand was approximated using 6 boxes, one for the palm and one for each of the fingers. In order to detect a collision we check if each one of the points constituting the point cloud is inside the boxes of the hand. One example of collision-free hand poses generated for the cup can be observed in Fig. 5.

Object	Old method (%)	MVBB method (%)
Colander	5.25	61.66
Cup	7.5	52.5
Plate	28	80
Pot	34.5	97.5
<b>Average</b>	<b>18.81</b>	<b>77.61</b>

TABLE I

COMPARISON OF THE SUCCESSFUL GRASP PERCENTAGE OF THE METHOD PRESENTED IN [1] AND THE PORPOSED IN THIS PAPER

## IV. SIMULATIONS

In this section, we present the simulations preformed for the set of kitchenware objects presented in Fig. 7, which are extracted from the PaCMan Grasp Dataset available at [13]. This figure shows: a) the mesh of the object used for simulations and for MVBB generation, b) the first 2 MVBBs generated by Algorithm 1, c) a finer MVBB generation, d) the first MVBB selected for the method, and finally d) show the first hand pose suggested by the method proposed in this paper.

Table I reports the comparison of the percentage of successful grasp generated with the method presented in this paper and the one presented in [1]. The results are improved mainly because of two reasons, 1) simulations never fail because of collision of the hand and the object at the beginning of the simulations, and 2) the MVBB decomposition helps to better place the hand with respect to the object, also giving more informed options about where and how the variations must be generated.

In order to go through all the process we take the example of the kettle. This is an object that, in practice, because of the dimensions of the object itself and those of the hand, can be grasped just either from the handle or the spout. The first step is to decompose the object in MVBBs. As it can be observed in Fig. 9(b) the algorithm presented in Section II segments the handle and the spout. The next step is to select a box to grasp: Fig. 9(c) shows the selected box in red — this is the outermost one. Then, the collision free hand poses are generated for the selected poses, see Fig. 9(d). Finally, the simulations are performed. Fig 9(e) shows the

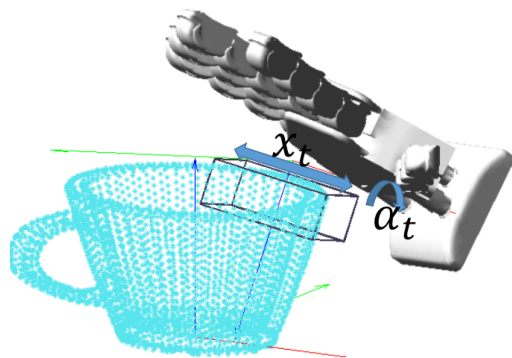


Fig. 6. In order to generate more poses to grasp each box, the hand is rotated and translated along  $x$  axis of the box.

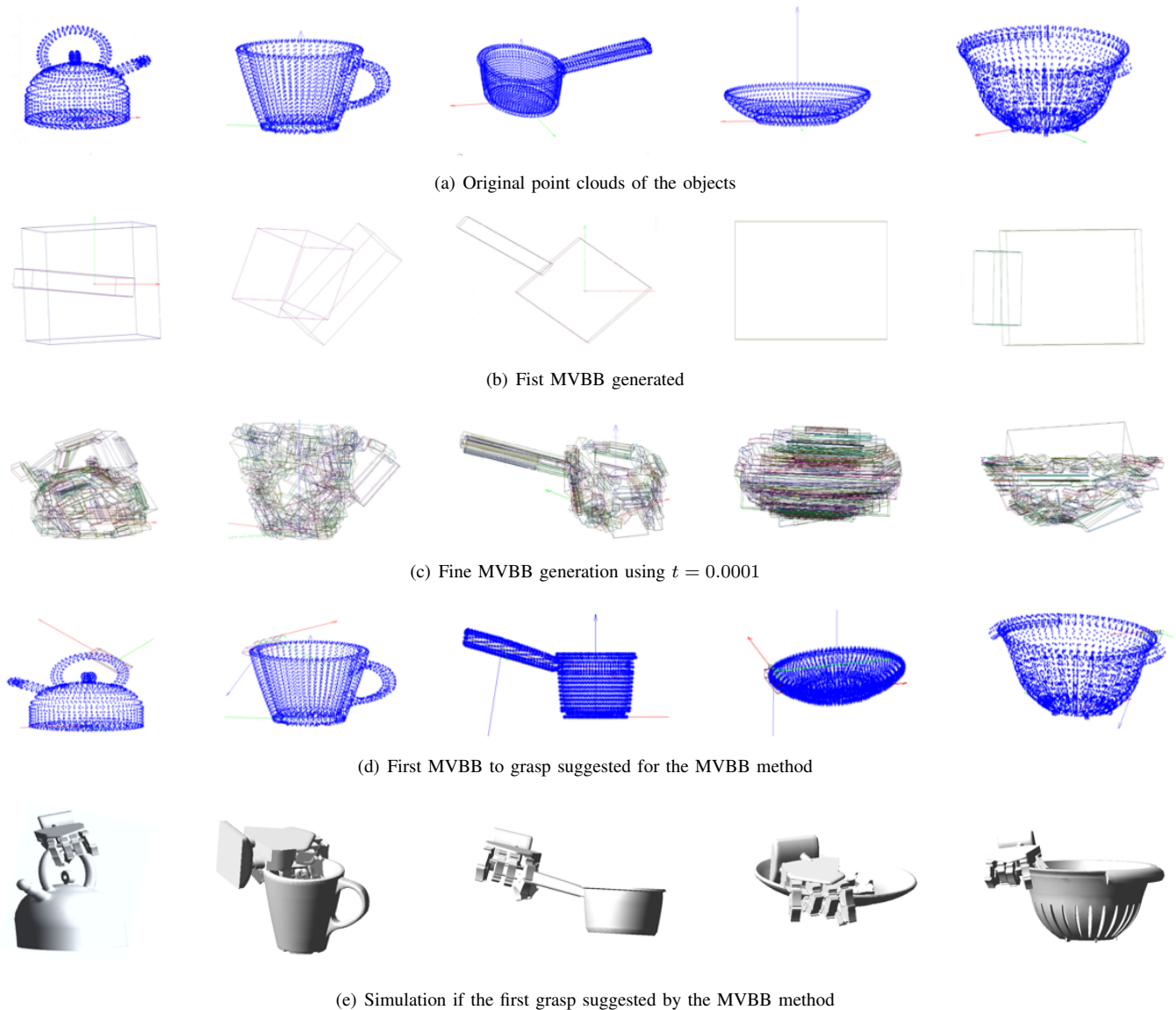


Fig. 7. Set of objects used for simulations

final configuration of the hand in of one of the simulations where the box-guided grasp suggestion results in the hand grasping the kettle from the handle.

#### A. Observations

Form the simulations performed in this work we observe that most of the failed simulations were caused by the hand being fixed during the whole grasping procedure. Observing how humans grasp an object, we can affirm that they normally adapt the position of the hand with respect to the object to avoid the object being ejected. This suggests to implement a strategy to try to reproduce the same behavior. This could be implemented observing the evolution of the contact forces as the object moves.

Another point to take into account is how strongly to close the hand to grasp an object. In some simulations, as an example when grasping the plate, the hand closes too much.

Therefore, even when the grasp is successful the grasp is still not human-like, as the fingers are too wrinkled. See Fig. 10. A strategy to load the motor current in simulations is envisioned for future research.

Fig. 8 shows the  $x_t$  vs  $\alpha_t$  space. Green points represent the successful grasp and red points the unsuccessful ones for experiment with the kettle. This plot suggest that a second phase of the procedure can be included to bias new configurations to the regions where there exist more successful grasp. The size of this region can be considered also as a measurement of the grasp quality and robustness of the box.

## V. CONCLUSIONS

Inspired by the new developments in robotic hands, in this paper we presented an algorithm to propose hand poses for grasp a objects. The algorithm consists in, firstly approxi-

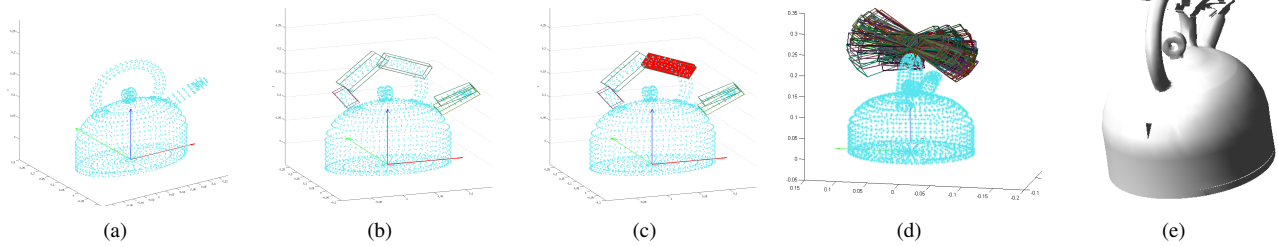


Fig. 9. Grasp planning using the method presented in this paper was executed for the kettle.

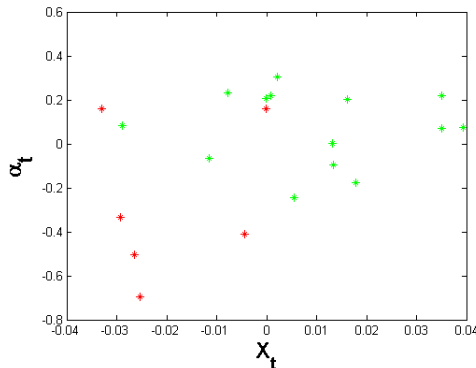


Fig. 8. 2D Space representing all possible variations for a box. Green points are successful grasps while red unsuccessful.

mating the object with a set of MVBB. Then, as the main contribution of this paper, we present a method to select a MVBB in the object and then align the hand. Performing dynamic simulations of the PISA/IIT soft hand we register, with the proposed algorithm, an increase in the probability of success of a proposed pose from 18.1% to 77.61% with respect to the method presented in [1].

Applying the algorithm presented in this paper to a point cloud coming from a 3D sensor and trying to generate hand postures in real-time is left for near future work.

#### ACKNOWLEDGMENTS

This work is supported by the EC under the CP-IP grant no. 600918 “PaCMan”, within the FP7-ICT-2011-9 program “Cognitive Systems”, ERC Advanced Grant no.

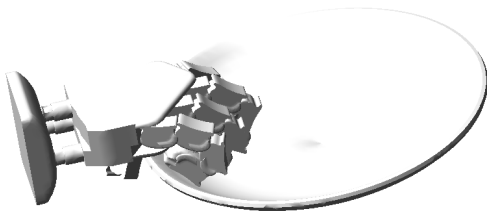


Fig. 10. This figure shows non human like but successful grasp. This problem is generated because the hand is always closed to the maximum.

291166 “SoftHands” - A Theory of Soft Synergies for a New Generation of Artificial Hands-, under grant agreements no.611832 “Walk-Man” and by CONACYT through the scholarship 266745/215873.

#### REFERENCES

- [1] M. Bonilla, E. Farnioli, C. Piazza, M. G. Catalano, G. Grioli, M. Garabini, M. Gabiccini, and A. Bicchi, “Grasping with soft hands,” in *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, Madrid, Spain, November 18 - 20, In Press.
- [2] K. Shimoga, “Robot grasp synthesis algorithms: A survey,” *Int. Journal of Robotics Research (IJRR)*, vol. 15, no. 3, pp. 230–266, 1996.
- [3] A. T. Miller, S. Knoop, H. Christensen, and P. Allen, “Automatic grasp planning using shape primitives,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003.
- [4] C. Borst, M. Fischer, and G. Hirzinger, “Grasp planning: How to choose a suitable task wrench space,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.
- [5] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelosoff, “Grasp planning via decomposition trees,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2007.
- [6] G. Biegelbauer and M. Vincze, “Efficient 3d object detection by fitting superquadrics to range image data for robots object manipulation,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2007.
- [7] L. Chevalier, F. Jaillet, and A. Baskurt, “Segmentation and superquadric modeling of 3d objects,” *Journal of WSCG*, vol. 11, no. 1, pp. 1–8, 2003.
- [8] K. Huebner, S. Ruthotto, and D. Kragic, “Minimum volume bounding box decomposition for shape approximation in robot grasping,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2008, pp. 1628–1633.
- [9] S. Geidenstam, K. Huebner, D. Banksell, and D. Kragic, “Learning of 2D grasping strategies from box-based 3D object approximations,” in *Robotics: Science and Systems (RSS)*, Seattle, USA, June 2009.
- [10] K. Huebner and D. Kragic, “Selection of robot pre-grasps using box-based shape approximation,” in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Sept 2008, pp. 1765–1770.
- [11] M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, “Adaptive synergies for the design and control of the pisa/iit soft hand,” *Int. Journal of Robotics Research (IJRR)*, vol. 33, p. 768–782, 2014.
- [12] R. Deimel and O. Brock, “A novel type of compliant, underactuated robotic hand for dexterous grasping,” in *Robotics: Science and Systems (RSS)*, Berkeley, USA, July 2014.
- [13] “Pacman grasp dataset,” 2015. [Online]. Available: <https://github.com/CentroEPIaggio/unipi-grasp-datasets/tree/master/scenario1>
- [14] MSC Software. (2015) Adams. [Online]. Available: <http://web.mscsoftware.com/>