Grasping Bulky Objects with Two Anthropomorphic Hands

Abiud Rojas-de-Silva and Raúl Suárez

Abstract— **This paper presents an algorithm to compute precision grasps for bulky objects using two anthropomorphic hands. We use objects modeled as point clouds obtained from a sensor camera or from a CAD model. We then process the point clouds dividing them into two set of slices where we look for sets of triplets of points. Each triplet must accomplish some physical conditions based on the structure of the hands. Then, the triplets of points from each set of slices are evaluated to find a combination that satisfies the force closure condition (FC). Once one valid couple of triplets have been found the inverse kinematics of the system is computed in order to know if the corresponding points are reachable by the hands, if so, motion planning and a collision check are performed to asses if the final grasp configuration of the system is suitable. The paper includes some application examples of the proposed approach.**

I. INTRODUCTION

Thanks to continuous advances in technology, robots become physically more robust and capable to perform tasks that require the execution of multiple actions. Among the main tasks to be done by robots are the grasp and manipulation of objects. Fundamental issues relating to grasping and dexterous manipulation have been investigated over years in order to find suitable and stable grasps able to resist external disturbances and that, in turn, allow the manipulation of the object.

Moreover, anthropomorphic hands have been developed as versatile end-effector tools able to perform complex grasp and manipulation tasks. This has opened a broad field of research and the problem of finding suitable grasps reachable by a given robotic hand still has not been solved completely, furthermore, in the current works, the use of two hands is frequently limited to regrasp approaches.

Despite there are many works dealing with grasp planning, the problem of how to grasp big or bulky objects using anthropomorphic systems has not been explored enough. This work deals with the problem of how to grasp bulky objects -that cannot be grasped with only one hand- using two anthropomorphic hands simultaneously, and therefore considering the constraints of the hands and arms in the search of the contact points on the surface of the object.

After this introduction the paper is organized as follows. Section II presents a review of related works and Section III presents the preliminaries of the work. Section IV presents an overview of the proposed approach, and Section V presents the details of the proposed algorithm. Then, Section VI shows some application examples and, finally, Section VII summarizes the work and presents some topics deserving future work.

II. RELATED WORKS

Finding a suitable grasp among the infinite possible candidates is a challenging problem that has been addressed frequently in the robotics community. Grasp planning is a field of research that has been studied for years and many approaches have been proposed to find solutions with a low computational cost.

One branch of research is focused on grasping at the contact level, where the main objective is to find the location of the contact points on the object regardless of hand kinematics. This area of research have been widely used in the synthesis of grasps for 2D [1], [2], [3] as well as for 3D objects [4], [5], [6]. Moreover, the concept of independent contact regions (ICRs) was introduced in order to provide robustness to the grasp in front of finger positioning errors [7], ensuring force-closure grasps independently of the exact position of the fingers within the ICRs. This concept has been developed by several researchers for 2D [8], [9] and 3D objects [10], [4], [11], [6] without considering whether they are reachable by an specific hand.

On the other hand, there are works addressing the problem of grasp planning taking into account the hand kinematics, providing the contact points and the corresponding hand pose simultaneously. Many of this approaches use optimization methods to find valid hand configurations [12], [13], [14], [15], [16], or discretize the workspace of the hands in order to simplify the search of suitable poses of the fingers on the object surface [17], [18], [19], or select the best grasp using a set of grasp quality metrics [20], [21].

Another important branch of research is the bimanual grasp and manipulation, where both robots interact with the same object. Several approaches and applications can be found in the literature [22]. However, the search of reachable grasps for bulky objects using two hands is a problem that has not been completely studied and the literature about this problem is reduced. Among the most relevant works in this line there is one that determines if two arbitrary contacts from each of two hands on an 2D unknown object yield a force closure grasp [23]. In contrast to this work, we present an algorithm focused on the search of reachable contact points.

Another relevant work presents the development of a bimanual grasp planner for cluttered environments that considers the two hands of the robot system as two fingers

The authors are with the Institute of Industrial and Control Engineering (IOC) at Universitat Politècnica de Catalunya (UPC), Barcelona, Spain (francisco.abiud.rojas.de.silva, raul.suarez@upc.edu). This work was partially supported by the Spanish Government through the projects DPI2013-40882-P and DPI2014-57757-R. Abiud Rojas-de-Silva was partially supported by the Mexican CONACyT doctoral grant 313768.

connected to a virtual wrist by means of virtual joints, turning the problem into a single-grasp problem, then a cost function is used to determine a suitable grasp pose of the hand based on a given preshape, then, the hand is closed at each iteration in order to find the contacts points that eventually yield a force closure grasp [24]. Unlike to this work we present an algorithm to search the contact points based on physical parameters of the hand, that allow to find reachable grasps with a reduced computational time in comparison with the discussed approach.

Another bimanual grasp planner for solid polyhedra was developed considering that polyhedron without parallel faces can be immobilized with only three contacts that can be located on any face, edge or vertex of the object. The potential contact points are classified based on its location, then the robot uses this classification to decide which part of its body must be used to touch a specific contact point. Hence, the robot can grasp the object using its torso and any link, joint or end-effector of both arms [25]. This approach uses the whole body of the robot to immobilize large objects, unlike to our work which is focused on finding contact points just for two anthropomorphic hands.

A different motion planning approach for dual arm manipulation and regrasping chooses a suitable grasp for one hand from a predefined set of grasps, then the corresponding joint configuration of the robot is computed and finally the object is moved to a suitable position to allow the grasping by the second hand [26]. As an extension of such approach, the authors developed a bimanual grasp planner where the position of one hand is computed selecting a random grasp obtained from a generator grasp candidates, based on an specific object representation called *the grip medial sphere* [27]. Then the position of the other hand is computed by selecting the grasp candidates that lies into a reachability space, previously computed. An inverse kinematics query for both end-effectors is computed and if there is a solution, the hands are closed until they touch the object and, finally, it is verified whether the grasp satisfies the force closure condition [28]. These two approaches have a common feature, the pose of the hands are not computed simultaneously, different to our work, furthermore the former approach uses predefined grasps for the objects, focusing mainly in the motion planning problem. In the second approach a grasp candidate generator is used, which based on the results reported in [27] the computational cost to find a grasp candidate (0.5 s per candidate, without taking into account the time employed for the object representation process), is higher than in our approach. There are other grasp candidate generators that only differs in the method to select the approach direction of the hand, using for instance the minimum volume bounding box [29] or the ray sampling strategy [30], but they share the same philosophy that consist in positioning the hand and then close it to find some contact points that can eventually yield a force closure grasps. Although these grasp generators can be used to search for bimanual grasp, they are designed to compute grasps for each hand independently, different to our method, which is able to compute a grasp using the two hands simultaneously. A grasp planner that is able to compute simultaneous grasps for two hands was developed based on the computation of grasps hypothesis during the tree generation of a motion planner [31], however, similar to the approaches previously mentioned, it computes the contact points after closing the hands and the computation time reported is higher than ours.

III. PRELIMINARIES

A. Object Model

The object surfaces are represented by a 3D point clouds (PC), which can be incomplete, but their density must be large enough to accurately represent the objects. The object models can be obtained either by a 3D vision system like the Kinect sensor of Microsoft or from a CAD model. The surface normal direction at each point was computed using the least-square plane fitting estimation method [32]. All the objects are bulky and cannot be grasped with only one hand, thus, two hands are needed to grasp them.

B. Number of fingers per hand

We only consider precision grasp with frictional contacts (i.e. only the fingertips are used), using the Coulomb's friction model to describe the frictional behavior between each finger and the object.

As we can find in the literature [33], using frictional contact points 4 fingers are sufficient to get a force closure (FC) grasp for 3D objects, regardless of the friction coefficient. Based on this, we decided to use 2 fingers and the thumb from each hand (i.e. two contacts more than the minimum needed) for the following reasons:

1) Adding one finger per hand (6 contacts in total) can significantly increase the robustness of the grasp.
2) A more human-like grasp may be obtained.
3) The approach can be easily used with other hands (anthropomorphic or not) with three fingers.

On the other hand, including the fourth finger of the hands increases significantly the time for computing the inverse kinematics. However, the effect of this drawback could be reduced by using faster methods to solve the inverse kinematics [34], [35].

C. Hardware and software tools

The hardware used in this work is the dual arm system ADARS (Anthropomorphic Dual Arm Robotic System), which is composed of two arms (UR5) with 6 DOF from Universal Robots, and each one is equipped with a four-finger Allegro Hand (AH) with 4 DOF per finger (see Fig. 1).

The collision checking, motion planning and simulations have been performed using *The Kautham Project* [36], which is a home-developed open source environment that provides several useful tools for the development of planners, like, for instance, random and deterministic sampling methods, metrics to evaluate the performance of planners and a simulation environments.

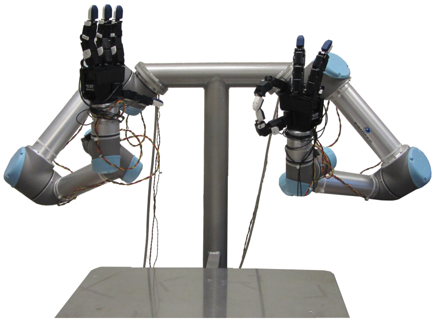

Fig. 1: ADARS: Athropomorphic Dual Arm Robotic System.

The point clouds has been managed using the Point Cloud Library (PCL) which is a large scale, open project for 2D/3D image and point cloud processing [37].

IV. PROBLEM STATEMENT AND APPROACH OVERVIEW

Grasping tasks have a great variety of specific problems, such as modeling the objects to be manipulated, determining how the object should to be grasped, verifying whether the grasp configuration is reachable by a given hand, planning the robot motions to reach the grasping position, among others. Additionally, as stated in Section II, most of the applications of dual arm systems are focused on the manipulation and regrasping of small objects and only a few of them address the problem of systematically determining grasps of large objects with two hands.

The aim of this work is to introduce a new method to determine grasps of bulky objects, modeled as point clouds, with two anthropomorphic hands.

The method consist in finding three reachable contact points for each hand. The search of the contact points is performed in an iterative way. First, the point cloud (PC) that describes the object surface is divided into two sets of slices SS_h (where $h = r, l$), one for each hand. In each slice S_i^h a set of triplets $ST_i^h = \{T_{i,j}^h, j = 1, ..., n\}$ is computed, with each triplet being composed of three points and their respective normal directions, $T_{i,j}^h = (\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3,\boldsymbol{n}_1,\boldsymbol{n}_2,\boldsymbol{n}_3)$. Then each triplet of ST_i^r is tested with each triplet of ST_i^l in order to find a couple of triplets $G = (T_{i,j}^r,\, T_{i,j}^l)$ that satisfies the FC condition with a quality above a given threshold. The grasp quality is evaluated as the largest perturbation that a grasp can resist in any direction [38]. Finally, a reachability analysis for the couple of triplets is performed by computing the inverse kinematics of the whole system, if there is a kinematic solution a motion planing and a collisions check are performed to ensure that the configuration of the whole system is suitable.

V. GRASPING FRAMEWORK

The proposed method is described in Algorithm 1, which is explained in the following subsections.

A. Approach direction

It has been shown that humans tends to align their hands with the main axis of inertia of the object before grasping it [39]. Based on this information, an *Approach direction* ($\boldsymbol{a}$) for both hands has been defined using one axis of inertia of the object.

Algorithm 1: Main Grasp Algorithm

```
input : $PC, \boldsymbol{v}, d$
output: $G$
1  $N \leftarrow \emptyset$
2  $p = false$
3  $N \leftarrow NormalEstimation(PC)$
4  $\{Axes,\, \boldsymbol{O}\} \leftarrow$ PCA $(PC)$
5  while $Axes \neq \emptyset$ do
6      $\boldsymbol{a} \leftarrow AxisApproach(Axes,\, \boldsymbol{v})$
7      $\Pi \leftarrow CompPlane(\boldsymbol{a})$
8      $\{SS_r,\, SS_l\} \leftarrow CompSlices(PC,\, N,\, \boldsymbol{a},\, \Pi,\, d)$
9      for Each $SS_h$ do
10         for Each $S_i^h \in SS_h$ do
11             $ST_i^h \leftarrow PotencialGrasp(S_i^h)$
12     $G \leftarrow findgrasp(ST^r,\, ST^l)$
13     if $G = \emptyset$ then
14         $Delete\ \boldsymbol{a}\ from\ Axes$
15         $\boldsymbol{a}.clear$
16     else
17         Return $G$
18 Grasp Not Found
```

To determine $\boldsymbol{a}$, we set a virtual axis $\boldsymbol{v}$, defined by the bases of the arms (see Fig. 2 top), then the axes of inertia of the object are computed with a principal component analysis (PCA) of the PC (line 4), that also returns the barycenter ($\boldsymbol{O}$) of the PC. Finally, we set as $\boldsymbol{a}$ the axis of inertia with the smaller angle with respect to $\boldsymbol{v}$ (line 6).

B. Object slices

Once the approach direction $\boldsymbol{a}$ has been computed, the object is divided into two sets of slices SS_r and SS_l, one for each hand using the function *CompSlices* (line 8). This function has as input a plane built using the point-normal representation,

$$\boldsymbol{n} \cdot (\boldsymbol{r} - \boldsymbol{r_0}) = 0 \qquad (1)$$

where the approach direction axis $\boldsymbol{a}$ is used as the normal vector $\mathbf{n}$ to the plane and $\boldsymbol{r_0}$ is the position vector of the farthest point of the object from the origin following the direction of $\boldsymbol{a}$. It means that the plane is located at one extreme of the object and it is orthogonal to $\boldsymbol{a}$ (Fig. 2 bottom). Then, the plane is displaced a distance d iteratively towards $\boldsymbol{O}$. At each iteration, the points lying below a given threshold distance from the plane are selected and saved as a slice S_i^h, until reaching the maximum number of slices *MaxSlice*, defined by the length of the fingers. When *MaxSlice* is reached, all the slices are saved as a set of slices SS_h and the process is repeated at the other extreme of the object obtaining the second set of slices (see Fig. 3).

C. Triplets extraction

The triplets extraction method is based on three physical conditions, similar to [16], but unlike this work, we do not use an optimization method. Instead we use the normal direction of the points as a filter to select only those points of

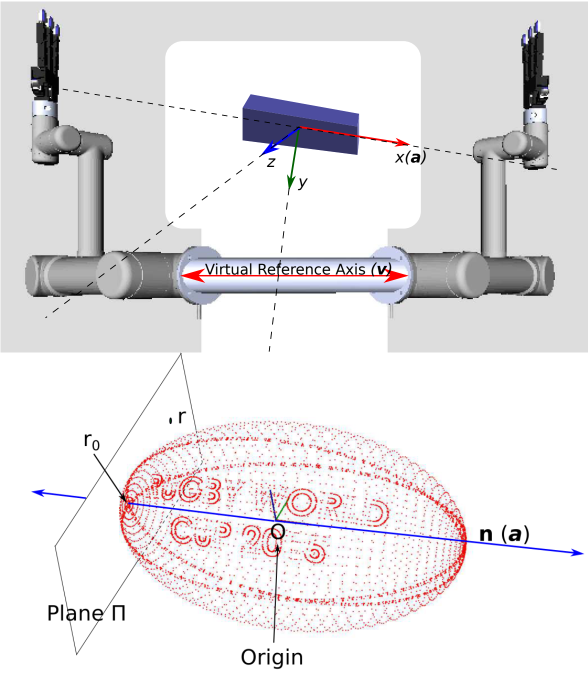

Fig. 2: Top: Location of the reference axis $\boldsymbol{v}$ in the robot system and definition of the X axis of the object as the axis $\boldsymbol{a}$. Bottom: Location of the plane Π on the object using the axis $\boldsymbol{a}$ as normal $\mathbf{n}$ to the plane.

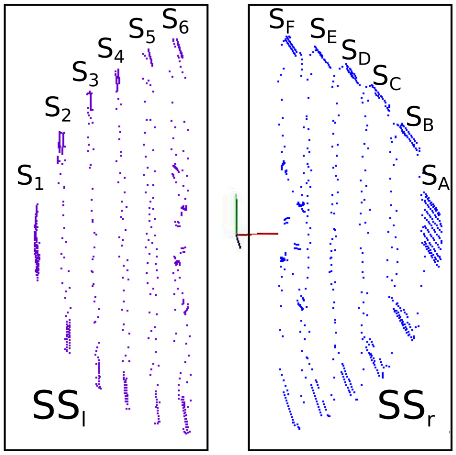

Fig. 3: Set of slices at each extreme of the object (each rectangle represents a set of slices).

each slice that can be candidates to form a grasp triplet, and then we select the triplets based on the following conditions:

- The resultant area of the triangle formed by the three points forming the triplet $A(T^h_{i,j})$, must be less than the maximum resultant area of the triangle formed by the center point of the three fingertips employed to grasp the object with the hand fully opened A_{hand}.

$$A(T^h_{i,j}) \leq A_{hand} \tag{2}$$

- The distance between the points of the triplet should be within the range

$$d_{min} \leq d_{ij} \leq d_{max} \tag{3}$$

being d_{min} the minimum distance between any pair of fingertips (close to zero), d_{ij} the distance between the contact points $\boldsymbol{p}_i$, $\boldsymbol{p}_j$ and d_{max} is the maximum allowed distance between the fingertips f_i, f_j.

- A quality measure is used to evaluate how close a triangle is to be an equilateral triangle, or, how to close is to be a degenerated triangle. The quality measure is

$$Q = \frac{(b+c-a)(c+a-b)(a+b-c)}{abc} \tag{4}$$

being a, b and c the lengths of the triangle sides. A quality close to zero means that the euclidean distance between the points that define one of its sides is close to zero or the three points are collinear. Each triplet must accomplish

$$Q(T^h_{i,j}) \geq mQ \tag{5}$$

where mQ is the minimum quality that a triangle must have to be considered acceptable.

Algorithm 2: Potencial Grasps

```
input : $S$
output: $ST$
1  $c0 = 0$
2  $n_{max}$
3  $\boldsymbol{S_t} \leftarrow PointsforThumb(S)$
4  $\boldsymbol{S_f} \leftarrow PointsforFingers(S)$
5  if $(\boldsymbol{S_t}.size > 0)$ and $(\boldsymbol{S_f}.size >= 2)$ then
6      $r = \boldsymbol{S_f}$.size
7      $nt = (\boldsymbol{S_t}.\text{size})\binom{2}{r}$
8      if $nt > nmax$ then
9          $nt = nmax$
10 else
11     Return $ST$
12 while $c0 < nt$ do
13     $tbp = randompoint(\boldsymbol{S_t})$
14     $ffip = randompoint(\boldsymbol{S_f})$
15     $sfp = randompoint(\boldsymbol{S_f})$
16     $ID = setID(tbp,ffip,sfp)$
17     $ids = ides \cup ID$
18     if $CheckId(ids, ID) = false$ then
19         $dTI = distance(tbp,\ ffip)$
20         $dIM = distance(ffip,\ sfp)$
21         $dMT = distance(sfp,\ tbp)$
22         if $Checkdistances(dTI,dIM,dMT) = false$ then
23             $A = A(dTI,\ dIM,\ dMT)$
24             $Q = Q(dTI,\ dIM,\ dMT)$
25             if $(A \leq MA)$ and $(Q \geq mQ)$ then
26                 $T \leftarrow fillT(tbp,\ ffip,\ sfp)$
27                 $ST \leftarrow ST \cup T$
28     $c0$++
29 Return $ST$
```

The triplets extraction from each slice is performed with the *PotencialGrasp* function (line 11) which is described in Algorithm 2 and it is roughly explained as follows:

- The functions *PointsforThumb* and *PointsforFingers* (lines 3 and 4) use the normal information of each point to classify them in two types: points whose normal

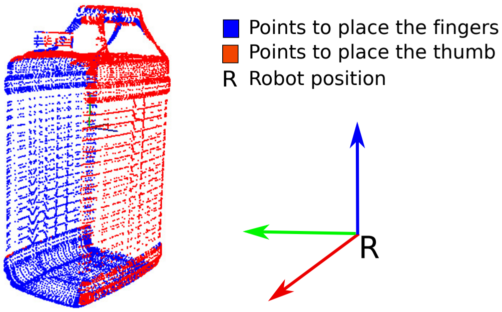

Fig. 4: Set of candidate points, blue points are selected to place the index and medium finger, red points are selected to place the thumb finger.

vectors point in opposite direction of the robot position are used to place the thumb $\boldsymbol{S_t}$, and points whose normal vectors point toward the robot are used to place the other two fingers $\boldsymbol{S_f}$ (See Fig 4), reducing considerably the number of points to analyze while looking for triplets.

- Once the points have been classified, it is verified whether there are enough points to form at least one triplet (line 5), and the number of triplets nt that can be found is computed (line 7) using the combinatorial expression $r \cdot \binom{2}{q}$ where r and q are the number of points in $\boldsymbol{S_t}$ and $\boldsymbol{S_f}$ respectively, then if the resultant number of candidates is higher than a maximum number of candidates per slice $nmax$, we set $nmax = nt$ (line 8-9).
- In the main loop three points are randomly selected, one from $\boldsymbol{S_t}$ and two from $\boldsymbol{S_f}$, an ID is assigned to the candidate triplet in order to verify if this combination has not been formed before (line 13-18). Then, the distances between the fingers are computed and the condition in eq.(3) is checked (line 19-22). If the distances accomplishes such condition, the area of the triangle A, formed by the three points, and its quality Q are computed (lines 23-24).
- Finally, if the area and the quality of the triangle accomplish the conditions in eq. (2) and (4), the three points are stored as a triplet $T^h_{i,j}$ in ST^h_i corresponding to the set of triples of the slice S^h_i (lines 26-27).

D. Finding the grasp

In order to find a grasp, every single triplet from one hand $T^r_{i,j}$ is evaluated along with every single triplet from the other hand $T^l_{i,j}$ to verify whether there is a couple of triplets $G = (T^r_{i,j}, T^l_{i,j})$ that accomplish the FC condition.

To find this couple of triplets the function *findgrasp* in Algorithm 1 (line 13) is used, which selects the sets of triplets following a specific strategy (see Fig. 5) to evaluate the triplets in an uniform way, trying that the grasping points are at the same level of slices for each hand.

Finally, the evaluation of the couples of triplets is performed with the function *CompGrasp* (Algorithm 3) which is inside of the *findgrasp* function. This function receives two set of triplets ST^h_i, and one triplet from each set is tested together until one couple accomplishes the FC condition (line 7) with a quality threshold, then the inverse kinematics of the whole system is computed in order to know whether the couple of triplets is reachable (line 9), if the couple is reachable, the motion planning and the collision check are performed (line 11). If a free-collision path for the whole system is found, the couple of triplets is returned as the solution grasp. In case that the inverse kinematics or the motion planning does not find a solution, the algorithm tries with another couple of triplets until a solution is found or until the last triplet of the last slice has been evaluated. In case that a valid grasp is not found, the algorithm re-starts using the next inertial axis with the smaller angle respect to $\boldsymbol{v}$. If after exploring the object through its three axis no grasp is found, the algorithm ends returning an error.

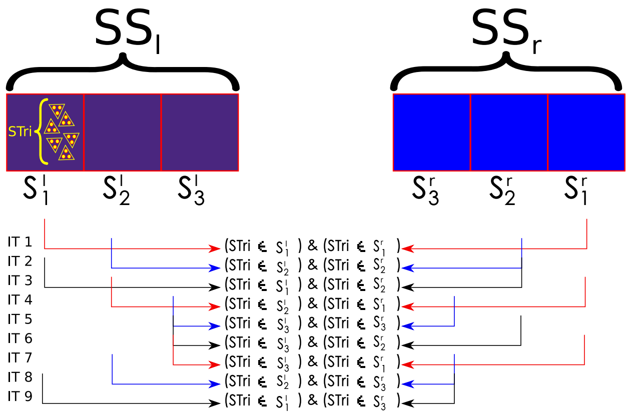
SS_l
SS_r
STri
S^l_1
S^l_2
S^l_3
S^r_3
S^r_2
S^r_1
IT 1
IT 2
IT 3
IT 4
IT 5
IT 6
IT 7
IT 8
IT 9
(STri ∈ S^l_1) & (STri ∈ S^r_1)
(STri ∈ S^l_2) & (STri ∈ S^r_2)
(STri ∈ S^l_1) & (STri ∈ S^r_2)
(STri ∈ S^l_2) & (STri ∈ S^r_1)
(STri ∈ S^l_3) & (STri ∈ S^r_3)
(STri ∈ S^l_3) & (STri ∈ S^r_2)
(STri ∈ S^l_3) & (STri ∈ S^r_1)
(STri ∈ S^l_2) & (STri ∈ S^r_3)
(STri ∈ S^l_1) & (STri ∈ S^r_3)

Fig. 5: Selection strategy to evaluate the triples of each slices.

Algorithm 3: CompGrasp

```
input : $ST^r, ST^l$
output: $G$
1  $fc, ik = false$
2  $lc \leftarrow \emptyset$
3  $rc \leftarrow \emptyset$
4  $initc$
5  for $i = 1$; $i <$ size of $ST^r$, $i{+}{+}$ do
6      for $j = 1$; $j <$ size of $ST^l$, $j{+}{+}$ do
7          $fc = FCTest(T^r_i, T^l_j)$
8          if $fc = true$ then
9              $ik = inversekinematics(T^r_i, T^l_j, lc, rc)$
10             if $ik = true$ then
11                 if $MotionPlanner(initc, lc, rc) = CollisionFreeSolution$ then
12                     $G \leftarrow \{T^r_i, T^l_j\}$
13                     Return $G$
14                 else
15                     $ik = false$
16 Return $G$
```

VI. EXPERIMENTS

To illustrate the performance of the proposed approach, we used 6 object models from the GrabCAD repository [40], which offers a huge variety of objects of different sizes, and

TABLE I: Global performance of the heuristic algorithm for grasping bulky objects (FC: Force Closure; IK: Inverse Kinematics; FCP: Free-Collision Path)

Object	FC grasp	IK solution	F-CP solution
Angrybird	93.33%	89.23%	84.00%
Ironmanmask	90.00%	85.18%	91.00%
Detergent	90.00%	77.77%	85.00%
Detergent 2	86.66%	92.30%	83.33%
Wheel	96.66%	86.20%	92.00%
Rugbyball	96.66%	96.55%	89.28%
Football	83.33%	100%	92%
Pan	70%	95.23%	95%
Wood	86.66%	96.15%	92%
Package	73.33%	86.36%	100%

TABLE II: Time employed to treat the point clouds (milliseconds) (Ne: Normal estimation; DA: Direction Approach)

Object	No. points	Ne(ms)	DA(ms)	Slices(ms)
Angrybird	16858	27	14	22
Ironmanmask	19884	27	15	28
Detergent	47675	56	29	41
Detergent 2	30089	37	23	38
Wheel	11258	37	8	14
Rugbyball	12226	29	11	15
Football	79744	98	61	103
Pan	34194	45	26	52
Wood	12197	15	9	19
Package	22237	31	17	34

TABLE III: Time to compute the grasp, inverse kinematics and motion planning (milliseconds) (MP: Motion Planning)

Object	Grasp(ms)	IK(ms)	MP(ms)	Total(ms)
Angrybird	67	1172	117	1356
Ironmanmask	88	817	208	1113
Detergent	133	1990	281	2404
Detergent 2	111	2345	156	2382
Wheel	60	1356	189	1605
Rugbyball	58	1555	235	1848
Football	287	3651	459	2558
Pan	146	2785	623	2417
Wood	57	1598	288	2328
Package	111	2679	389	2382

4 objects from the YCB object data base [41], that have been obtained using a high-resolution scan.

It was assumed that each object was made of hard plastic or wood, thus the friction coefficient was established as 0.4 (the range of mechanizable plastics is between 0.08 and 0.6). On the other hand, we consider that a FC grasp is acceptable if its quality is higher than 0.03.

30 trials per objects were performed (300 trials in total) and it can be seen that grasps with a quality higher than 0.03 are obtained in the 86.66% of the cases, from which 90.49% has inverse kinematics solution and from these 90.33% has a free-collision path solution (See Table I).

The time employed in the treatment of the point cloud that includes the computation of the normal vectors, the estimation of the direction approach and the computation of the object slices depends of the number of points in the cloud. Table II presents the average time employed in milliseconds for each object.

Table III presents the average time employed to find the grasp, solve the inverse kinematics and run the motion planner. As it can be seen, computing the inverse kinematics is the most expensive process with an average of 1994 milliseconds. We are currently using the transpose method which is stable and works acceptably well with several end effectors. The motion planning and collision-chek are performed using the OMPL planner RRTConnet, which is integrated in *The Kautham Project*. It is worth notice that the time employed to find the grasp on the scan-based objects is slightly higher than on the CAD-based objects, due to the noise present in the point cloud and its density, however the algorithm is able to find a solution in a rasonable time. The stable grasp, the final configuration of the robot for each object and the real experimentation with two objects are shown in Fig. 6, Fig. 7 and Fig. 8 respectively.

VII. Conclusions and Future Works

In this work, an heuristic algorithm to grasp bulky objects with two anthropomorphic hands has been presented, with the objects modeled as 3D point clouds. The algorithm is able to work with complete or incomplete point clouds, either coming from a sensor/scaner or from a CAD model. The main contribution is the search method, which is based in spliting the object into two sets of slices, then in each slice look for a set of triplets considering certain physical conditions and using the normal vector directions to discard the unuseful points. Then the triplets from each slice of one set of slices are evaluated together with the triplets of the other set of slices. The evaluation consists in finding a couple of triplets that accomplishes the FC condition with a quality larger than a given threshold. Then, if a couple is found the inverse kinematics of the whole system is solved to verify that the couple is actually reachable, if the inverse kinematics has a solution a motion planning and a collision-check are performed to check whether the grasp is suitable and there are not collisions.

In the approach, the points that form a triplet always belong to the same slice, therefore the fingers will be always contacting in the same slice. A future work is to improve the search method to allow triplets taking points from different slices if necessary, and we also want to consider the use of a fourth finger from each hand.

References

[1] Y.-H. Liu, “Computing n-finger force-closure grasps on polygonal objects,” in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3, May 1998, pp. 2734–2739 vol.3.

[2] N. Niparnan and A. Sudsang, “Computing All Force-Closure Grasps of 2D Objects from Contact Point Set,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct 2006, pp. 1599–1604.

[3] J. Cornella and R. Suarez, “Efficient Determination of Four-Point Form-Closure Optimal Constraints of Polygonal Objects,” *Automation Science and Engineering, IEEE Transactions on*, vol. 6, no. 1, pp. 121–130, Jan 2009.

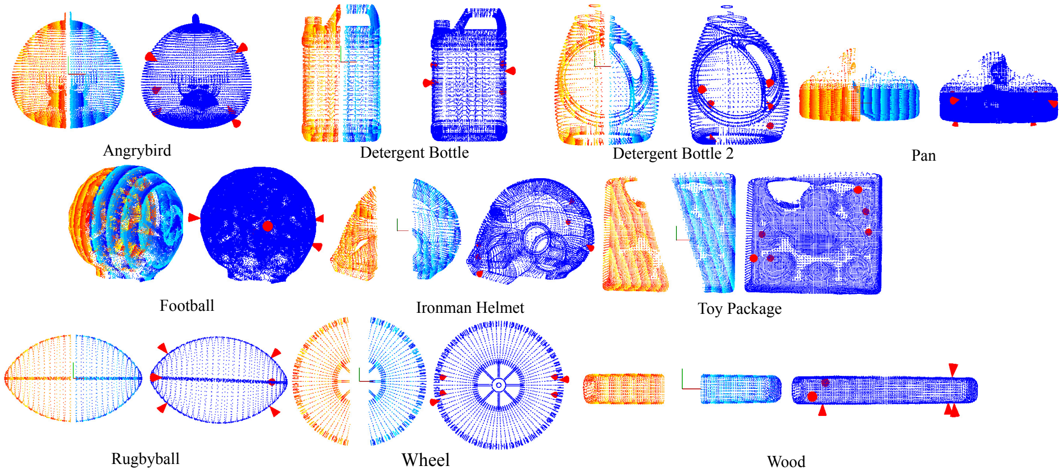

Fig. 6: Objects splited in slices based on their inertial axis and the resulting contact points (view from the robot side).

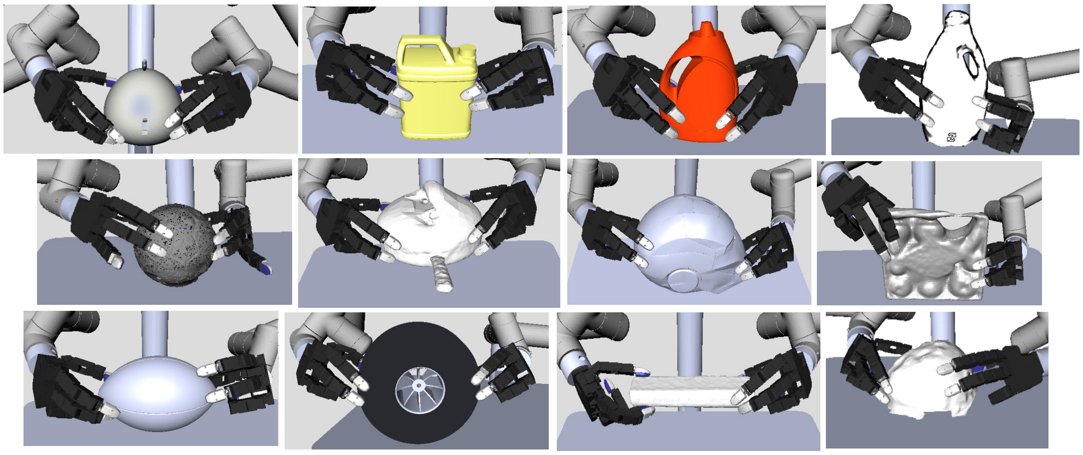

Fig. 7: Experimental simulation of grasp (frontal view).

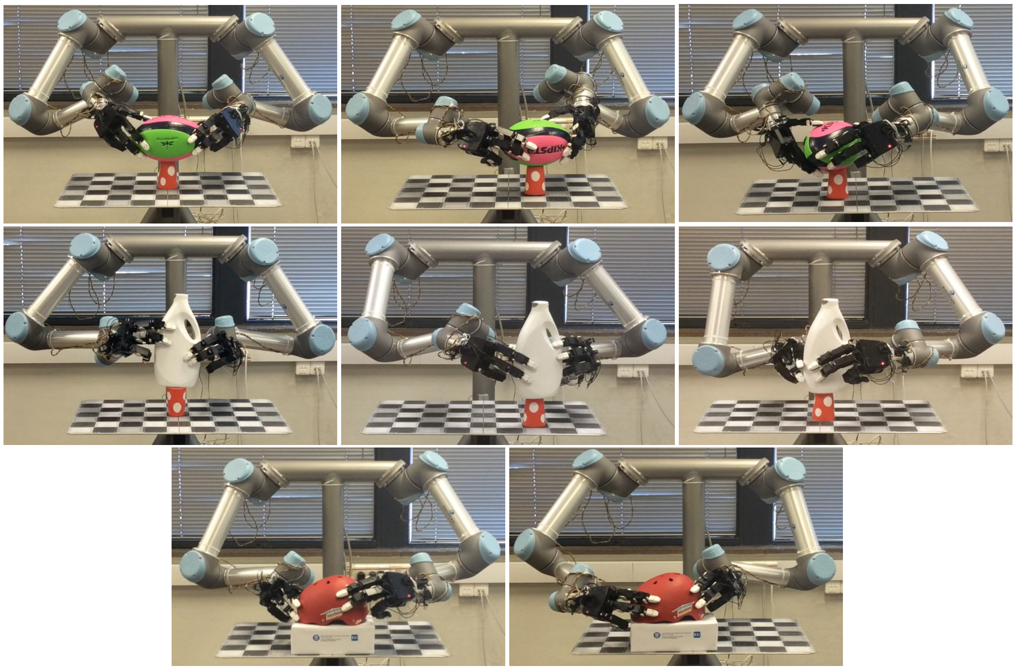

Fig. 8: Real experimentation with three objects, different setups and contact points.

[4] J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet, "On Computing Four-Finger Equilibrium and Force-Closure Grasps of Polyhedral Objects," *The International Journal of Robotics Research*, vol. 16, no. 1, pp. 11–35, 1997.

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

[6] M. A. Roa and R. Suarez, "Finding locally optimum force-closure grasp." in *Robotics and Computer-Integrated Manufacturing*, 2009.

[7] V.-D. Nguyen, *Constructing force-closure grasps*. Institute of Electrical and Electronics Engineers, 1988.

[8] J. Cornellà and R. Suárez, "Fast and Flexible Determination of Force-Closure Independent Regions to Grasp Polygonal Objects." in *ICRA*. IEEE, 2005, pp. 766–771.

[9] J. Cornella and R. Suarez, "Determining independent grasp regions on 2D discrete objects," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, Aug 2005, pp. 2941–2946.

[10] J. Ponce and B. Faverjon, "On computing three-finger force-closure grasps of polygonal objects," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 6, pp. 868–881, Dec 1995.

[11] N. S. Pollard, "Closure and Quality Equivalence for Efficient Synthesis of Grasps from Examples," *The International Journal of Robotics Research*, vol. 23, no. 6, pp. 595–613, 2004.

[12] C. Borst, M. Fischer, and G. Hirzinger, "Calculating hand configurations for precision and pinch grasps," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2, 2002, pp. 1553–1559 vol.2.

[13] Y. Guan and H. Zhang, "Kinematic feasibility analysis of 3D grasps," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3, 2001, pp. 2197–2202 vol.3.

[14] F. Gilart and R. Suarez, "Determining Force-Closure Grasps Reachable by a Given Hand," in *10th IFAC Symposium on Robot Control, SYROCO*, September 2012, pp. 235–240.

[15] C. Rosales, J. M. Porta, R. Suárez, and L. Ros, "Finding all valid hand configurations for a given precision grasp." in *ICRA*. IEEE, 2008, pp. 1634–1640.

[16] I. Gori, U. Pattacini, V. Tikhanoff, and G. Metta, "Three-finger precision grasp on incomplete 3d point clouds," in *Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA'14*, Hong Kong, China, 2014.

[17] Z. Xue, J. Marius Zoellner, and R. Dillmann, "Grasp planning: Find the contact points," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, Dec 2007, pp. 835–840.

[18] M. A. Roa, K. Hertkorn, C. Borst, and G. Hirzinger, "Reachable Independent Contact Regions for precision grasps." in *Robotics and Automation (ICRA), 2011 IEEE International Conference on IEEE International Conference on*, 2011, pp. 5337–5343.

[19] J.-P. Saut and D. Sidobre, "Efficient models for grasp planning with a multi-fingered hand." *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 347–357, 2012.

[20] M. Andrew and K. A. Peter, "Graspit!: A versatile simulator for robotic grasping," *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 110–112, 2004.

[21] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Autonomous Robots*, vol. 38, no. 1, pp. 65–88, 2014.

[22] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation - A survey." *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340–1353, 2012.

[23] J. Caraza and X. Yun, "Two-handed grasping with two-fingered hands," in *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, June 1991, pp. 597–602 vol.1.

[24] D. Berenson and S. Srinivasa, "Grasp synthesis in cluttered environments for dexterous hands," in *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, 2008.

[25] J. Seo and V. Kumar, "Spatial, bimanual, whole-arm grasping," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012.

[26] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Oct 2009, pp. 2464–2470.

[27] M. Przybylski, T. Asfour, and R. Dillmann, "Planning grasps for robotic hands using a novel object representation based on the medial axis transform," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, Sept 2011, pp. 1781–1788.

[28] N. Vahrenkamp, M. Przybylski, T. Asfour, and R. Dillmann, "Bimanual grasp planning." in *Humanoids*. IEEE, 2011, pp. 493–499.

[29] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008.

[30] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, 2010.

[31] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Simultaneous Grasp and Motion Planning," *IEEE Robotics and Automation Magazine*, vol. 19, pp. 43–57, 2012.

[32] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," PhD dissertation, Tecnische Universitatet Muenchen, October 2009.

[33] X. Markenscoff, L. Ni, and C. H. Papadimitriou, "The geometry of grasping," *International Journal of Robotics Research*, vol. 9, no. 1, pp. 61–74, 1990.

[34] S. R. Buss and J.-S. Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics Tools*, vol. 10, pp. 37–49, 2004.

[35] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," IEEE Journal of Robotics and Automation, Tech. Rep., 2004.

[36] J. Rosell, A. Pérez, A. Aliakbar, Muhayyuddin, L. Palomo, and N. García, "The kautham project: A teaching and research tool for robot motion planning," in *Proc. of the IEEE Int. Conf. on Emerging Technologies and Factory Automation, ETFA'14*, 2014. [Online]. Available: sir.upc.edu/kautham

[37] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *International Conference on Robotics and Automation*, Shanghai, China, 2011.

[38] C. Ferrari and J. Cany, "Planning optimal grasps," in *IEEE International Conference on Robotics and Automation*, 1992.

[39] R. Balasubramanian, L. Xu, P. Brook, J. Smith, and Y. Matsuoka, "Human-guided grasp measures improve grasp robustness on physical robot," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010.

[40] [Online]. Available: https://grabcad.com

[41] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. Dollar, "Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set," *Robotics Automation Magazine, IEEE*, vol. 22, no. 3, pp. 36–52, 2015.