Grasping Unknown Objects Based on Gripper Workspace Spheres

Mohamed Sorour, Khaled Elgeneidy, Aravinda Srinivasan, and Marc Hanheide

***Abstract*— In this paper, we present a novel grasp planning algorithm for unknown objects given a registered point cloud of the target from different views. The proposed methodology requires no prior knowledge of the object, nor offline learning. In our approach, the gripper kinematic model is used to generate a point cloud of each finger workspace, which is then filled with spheres. At run-time, first the object is segmented, its major axis is computed, in a plane perpendicular to which, the main grasping action is constrained. The object is then uniformly sampled and scanned for various gripper poses that assure at least one object point is located in the workspace of each finger. In addition, collision checks with the object or the table are performed using computationally inexpensive gripper shape approximation. Our methodology is both time efficient (consumes less than 1.5 seconds in average) and versatile. Successful experiments have been conducted on a simple jaw gripper (Franka Panda gripper) as well as a complex, high Degree of Freedom (DoF) hand (Allegro hand).**

***Index Terms*— grasping, manipulation.**

I. Introduction

Object grasping problem is usually approached based on physical analysis (classical approach) [1]–[4], geometry [5]–[8], or machine learning (ML) [9]–[11]. The first requires sufficient knowledge about the object (shape, mass, material, ... etc.) and as such inconvenient for unknown object grasping. Whereas the last has gained huge momentum in the past decade thanks to its ability to model very complex systems and the advances in hardware computational power. However, ML approaches requires extensive offline processing and sufficiently large training data sets, and at the moment, generalization to unknown objects, versatility to different gripper structures as well as algorithm processing time [12] remains an open challenge. On the other hand geometry based approaches generally offers less computation time, with no sacrifice on generality or success rates, under which this work is categorized.

In [13], a set of contact points that fulfill certain geometric conditions are computed for unknown objects in point cloud, these are ranked to find the most stable grasp. Their algorithm computes grasping points in lower than 1 second duration, however is limited to 2 fingered grippers, and no data regarding grasping success rate is presented. Grasp planning of unknown objects from point cloud data is presented in [5], using geometric information to categorize objects into shape primitives, with predefined strategies for each. Success rate of 82% is achieved, however no computation time data was reported. This approach is similar to the pioneering work in [6], [14] with the later employing machine learning in grasp selection. In [15], similar approach is employed, more suitable for generalization, however, only simulations are provided with no computation time data. In [7], object shape reconstruction is performed online from successive image data, their method is general for different kinds of multi-fingered hands, however execution time is reported as being significant. While in [16], fast shape reconstruction algorithm is presented as means of improving grasping algorithms.

Another geometric approach is used to synthesize force balanced grasps in [17], the algorithm is described fast, with execution time below 4 seconds [12], [18], however no exact data is given, as well as being tailored for 2 fingered grippers. In [8], the smallest computation time can be found (34ms), where a grasp planner is designed to fit only a jaw gripper by searching for two parallel line segments in the object image. However their method doesn't take into account the 3D shape of object and thus is only suitable for simple regular objects with parallel surfaces. They reported grasp planning is done using only one image, which raises questions about how the object location is computed?

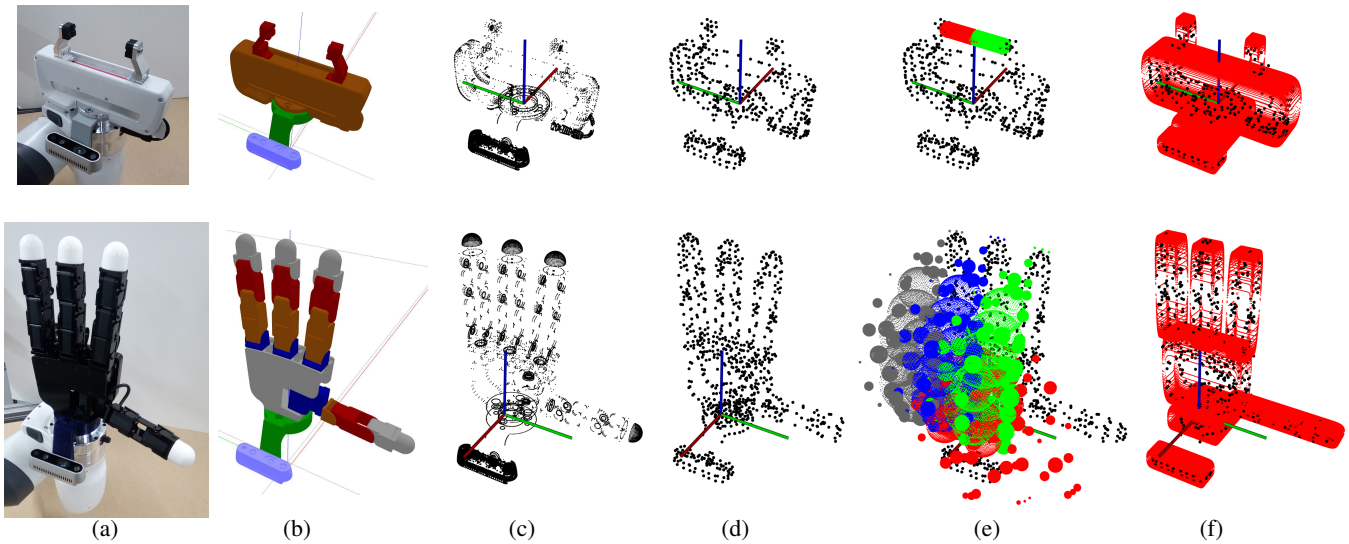
In [19], the authors presented a grasp planner using single depth image of a non-occluded object. Their work, however, is limited to 2 fingered grippers and the computation time varies highly up to 8 seconds. In [20], geometry based planner is implemented, execution time ranges within 2~3 seconds, however, is limited to parallel plate grippers.

Recently, the authors in [21] proposed a grasp planner based on similarity metric of local surface features between object and gripper's finger surfaces. Experiments on heap of objects were successfully conducted, however execution time is above 13 seconds using a 2 fingered gripper. The execution time is expected to multiply in case of employing multi-fingered hands which limits its use in real-time applications. Similar approach is presented in [22], with rather more freedom to modify gripper shape to match that of the object. However, computation time was not reported.

In this work, we introduce a novel algorithm for grasp planning, the input of which is a point cloud of an unknown object, while the output is the gripper pre-grasp pose in the form of homogeneous transformation matrix. In an offline step, a point cloud of the gripper workspace is generated using its kinematic model, then filled with spheres. At run-time, we register 3 point clouds of the scene taken from different angles to have better representation of the object, whose pose is computed, and its bounding box is sampled. These sampling points will be used for scanning the object using the gripper workspace spheres, by transforming the gripper workspace centroid to these points and iterating through several gripper orientations around the object major axis (along which, the object longest side exists). The objective is to find the best gripper pose, at which at least one object point is located inside each workspace finger as fast as possible, and hence comes the workspace spheres as a computationally affordable means of checking object points

Lincoln Center for Autonomous Systems (L-CAS), School of Computer Science, University of Lincoln, Brayford Pool, LN6 7TS Lincoln, United Kingdom. msorour@lincoln.ac.uk

Manuscript 1656 submitted to 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2019.



(a) (b) (c) (d) (e) (f)

Fig. 1: Gripper model offline processing, featuring the Franka-Emika panda gripper (upper) and the Allegro right hand (lower), each fitted with realsense D435 depth camera. Respectively from left to right: the real hardware, the CAD model, is then converted to a point cloud, which is then down sampled. Afterwards, the gripper workspace is sampled and filled with a predefined number of spheres. Lastly the gripper shape is approximated by a set of special ellipsoids.

being in or out the gripper's finger workspace.

In contrast to object shape approximating methods, here we use gripper approximation using few *special ellipsoids* (SE) as well as table plane approximation to simplify collision detection between grasp pose candidates and object/table. The contribution of our approach is twofold:

- Versatility: it can be applied to different gripper structures, we perform experiments featuring Franka Panda gripper (2 fingered) and Allegro right hand (4 fingered with 16 DoF).
- Computationally inexpensive: the algorithm can perform grasp planning in less than 0.7, 1.8 seconds for the parallel jaw and 4 fingered grippers respectively.

The paper is organised as follows, section II introduces the gripper model offline processing. The object major axis computation and sampling is briefed in section III. The grasping algorithm is detailed in section IV. Experiments and results are reported in section V. Conclusions are finally given in section VI.

II. The Gripper

In this section, we present the gripper model processing step of the grasping algorithm. This step is done once and offline, so its associated processing time does not add computation cost for the real-time execution. In the sequel we use extensively the notion of *special ellipsoid* for different purposes, this is a variation of the ellipsoid equation given by:

$$\frac{(x-x_0)^l}{a^l}+\frac{(y-y_0)^l}{b^l}+\frac{(z-z_0)^2}{c^2}=1, \tag{1}$$

where a, b, c are the principal semi axes of the ellipsoid, and x_0, y_0, z_0 denote the offset from origin. As the power l increases, better cuboid approximation is obtained, as depicted in Fig. 2. A compact form of the left hand side in (1) will be referred to in the sequel for convenience by:

$$EvalSE(\mathcal{E}_o, \mathcal{E}_p, \mathcal{C}, l), \tag{2}$$

where $\mathcal{E}_o$, $\mathcal{E}_p$, $\mathcal{C}$ denote the special ellipsoid offset and semi-principal vectors, and the cloud point(s) whose belonging to the SE parameterized by $\mathcal{E}_o$, $\mathcal{E}_p$ is to be evaluated respectively.

A summary of the gripper processing operations is shown in Fig. 1, where the gripper computer aided design in (CAD) model Fig. 1(b) is converted to a point cloud in Fig. 1(c), which is then downsampled in Fig. 1(d), these steps are shown for the Franka-Emika panda gripper [23] (upper) and the Allegro right hand [24] (lower). To this end, let ${}^g\mathcal{C}_g \in \mathbb{R}^{n_g \times 3}$ denote the downsampled gripper point cloud expressed in the gripper frame (Fig. 1(d)), where n_g is the number of cloud points.

Next, the direct geometric model (DGM) of the gripper is formulated, this relates the gripper finger tip pose to the joint space configuration. The DGM is simple in the case of parallel jaw gripper (identity matrix). However, it is complex for the allegro hand, for which we have 4 DGMs, one per finger. By sampling the joint space (each joint range is divided into equally spaced values) of each finger, we use the DGM to get the corresponding sampled operation space of each finger tip. The sampled operation space of each finger is then filled up with spheres, this is done by selecting a number of points inside the sampled operation space and use these as offsets for the set of spheres. The largest possible radius of sphere at each of these offsets is used such that each of the spheres remain entirely inside the workspace

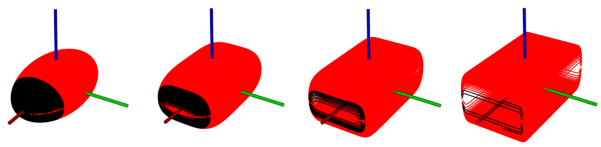


Fig. 2: Cuboid shape approximation using special ellipsoid in (1), with $l = 2, 4, 10, 30$ respectively from left to right.

Manuscript 1656 submitted to 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2019.

of the corresponding finger. Gripper workspace spheres are show in Fig. 1(e) (upper), for the Franka panda right and left fingers in red and green, and in Fig. 1(e) (lower) for the Allegro hand's thumb, index, middle, and pinky fingers in red, green, blue, and grey respectively. Let the set of gripper's finger workspace spheres ${}^{g}\mathcal{S}^{f} = \{{}^{g}\mathcal{S}^{f}_{o}, {}^{g}\mathcal{S}^{f}_{r}\}$ be defined by the set of sphere offsets ${}^{g}\mathcal{S}^{f}_{o} \in \mathbb{R}^{n_{sp}\times 3}$ and the set of sphere radii ${}^{g}\mathcal{S}^{f}_{r} \in \mathbb{R}^{n_{sp}\times 1}$, expressed in the gripper frame with $f \in \{1 \dots n_f\}$ denoting the gripper finger index, and n_{sp}, $n_f \geq 2$, the number of spheres and fingers respectively. Analogous to (2), a compact form for the left hand side of the sphere formula evaluation can be given by:

$$EvalSphere(\mathcal{S}_o, \mathcal{S}_r, \mathcal{C}), \tag{3}$$

where $\mathcal{S}_o$, $\mathcal{S}_r$, $\mathcal{C}$ denote the sphere offset vector and radius, and the cloud point(s) whose belonging to the sphere parameterized by $\mathcal{S}_o$, $\mathcal{S}_r$ is to be evaluated respectively.

Finally the gripper shape is approximated by a set of special ellipsoids (1) as shown in Fig. 1(f). These provide a computationally efficient means to check if a particular gripper pose collides with the object or not, simply by evaluating (1) for each object cloud point, for the set of gripper special ellipsoids defined by the set of offsets ${}^{g}\mathcal{E}_{go} \in \mathbb{R}^{n_e\times 3}$ and the set of semi principal axes ${}^{g}\mathcal{E}_{gp} \in \mathbb{R}^{n_e\times 3}$, expressed in the gripper frame, with n_e denoting the number of gripper special ellipsoids.

III. The Object

The scene observed by the depth camera is assumed to contain only one object, as shown in Fig. 3(a). A good point cloud representation of which is acquired via 3 view points, the relative transformation of which are accurately measured (since they are performed by moving the end effector of the Franka panda arm), and as such a single scene point cloud is reconstructed, as in Fig. 3(b). The point cloud is then downsampled in Fig. 3(c) to save computation time in further required processing.

The object is assumed to be placed on top of a planar surface (table), which is first segmented by finding all points that fit a plane model using random sample consensus (RANSAC) as a robust estimator. The object is then segmented using semantic 3D object models [25] implemented in point cloud library (PCL) [26].

A. Object Pose Computation and Sampling

In concept this work assumes a set of semantic rules, human inspired for successful grasping. The first is that an object should be approached in an orientation perpendicular to its major axis (the axis lies along the longest object dimension). As such we need to compute the object homogeneous transformation matrix with respect to the gripper frame, here we developed a simple algorithm to do that at a low computational cost. In addition to approximating the object's bounding box general dimensions which will be used in object sampling.

To do this we first compute the object's centroid point from the downsampled object cloud, search for the 2 farthest opposite points of the object, using these we divide the cloud into 2 point clouds based on their distance to the aforementioned far points. By computing the centroids of these 2 point clouds we construct the z-axis, along which is the longest object dimension. Then the obtained z-axis vector is projected onto the xy plane alongside the object's point cloud. The z-axis line divides the point cloud into 2 sets, the centroid of each is again computed, then used to obtain an orthogonal axis to the z-axis, which we arbitrarily designate it as the x-axis. The y-axis is then obtained by cross multiplication of the x and z axes to form the orthogonal object frame located at the object centroid. By projecting the normalized object frame onto that of the gripper we obtain the object transformation matrix expressed in gripper frame ${}^{g}\mathbf{T}_{o}$. Using this method, the obtained coordinate frames for the object and the table are shown in Fig. 3(d), with the larger frame being that of the object. To this end, we define ${}^{c}\mathcal{C}_{o} \in \mathbb{R}^{n_o\times 3}$ as the downsampled object point cloud expressed in the camera frame shown in Fig. 3(c), where n_o is the number of object cloud points. The camera frame is related to the arm's end effector frame by a static transformation matrix ${}^{e}\mathbf{T}_{c}$, acquired during camera calibration.

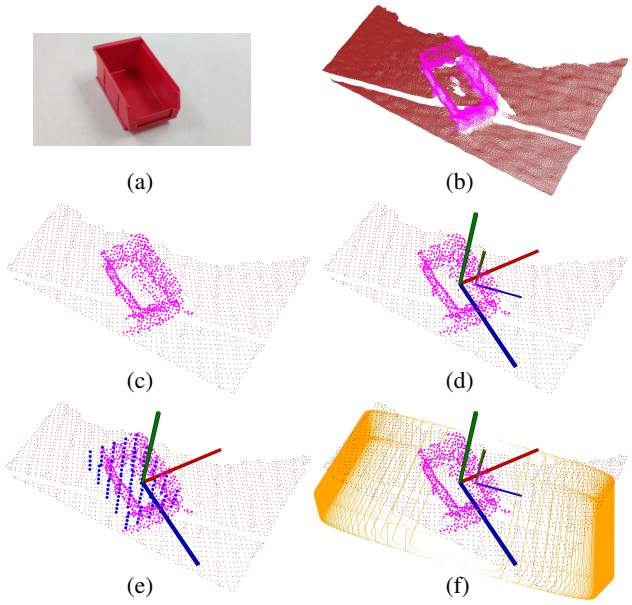


Fig. 3: Object point cloud processing, featuring the storage bin as the object in (a). Registered and segmented multi-view point cloud in (b) is then downsampled (c) and object/table coordinate frames (transformation matrices) are computed (d). The object's bounding box is sampled in (e), and the table special ellipsoid is constructed in (f).

Furthermore, We use the object's general dimensions obtained to sample the bounding box along the object's coordinate axes, as depicted by blue dots in Fig. 3(e), let ${}^{o}\mathcal{C}_{s} \in \mathbb{R}^{n_s\times 3}$ denote the point cloud of the sampled object bounding box, with n_s the number of sample points. These sample points will be used in scanning the object cloud for points that belong to the gripper workspace spheres (refer to Fig. 1) for a given gripper pose with respect to the object.

B. Table Special Ellipsoid

The table coordinate frame and transformation matrix ${}^{g}\mathbf{T}_{t}$ are used to construct the *table special ellipsoid*, colored orange in Fig. 3(f). This provides a computationally efficient means to detect gripper collision with table for a given gripper pose, by evaluating the table SE at each point of

the downsampled gripper cloud expressed in the table frame ${}^{t}\mathcal{C}_g = {}^{g}\mathbf{T}_t^{-1}{}^{g}\mathcal{C}_g$. It is defined by the $3D$ offset point ${}^{t}\mathcal{E}_{to} \in \mathbb{R}^{1\times 3}$ and the vector of semi principal axes ${}^{t}\mathcal{E}_{tp} \in \mathbb{R}^{1\times 3}$, expressed in the table frame.

IV. Grasping Algorithm

The output of the grasping algorithm provided in Algorithm 1 is the desired end effector transformation matrix expressed in its frame ${}^{e}\mathbf{T}_{e^*}$, which should be implemented by the robot arm to position the gripper in a good pose seeking a successful grasping. This pose is evaluated based on a simple set of human observed criteria. The input to the grasping algorithm is the set of point clouds, gripper workspace spheres and special ellipsoids parameterisation detailed in previous sections. We assume that the object is within the arm's workspace, and as such the output gripper pose can be realized using the arm.

Algorithm 1 Grasping pose algorithm

Input: downsampled object point cloud ${}^{c}\mathcal{C}_o$,
sampled "object bounding box" point cloud ${}^{o}\mathcal{C}_s$,
downsampled gripper point cloud ${}^{g}\mathcal{C}_g$,
gripper workspace spheres ${}^{g}\mathcal{S}_o^f$, ${}^{g}\mathcal{S}_r^f$,
gripper special ellipsoids ${}^{g}\mathcal{E}_{go}$, ${}^{g}\mathcal{E}_{gp}$,
table special ellipsoid ${}^{t}\mathcal{E}_{to}$, ${}^{t}\mathcal{E}_{tp}$
Output: desired effector transformation ${}^{e}\mathbf{T}_{e^*}$

```
g S_{i,j}^{f(*)} = ∅, g C̄_o^{i,j(*)} = ∅, d* = ∞
e T_gc = e T_g g T_o Rot(x, π/2)
for each point o c_s^i in o C_s do
    e T_gc^i = Trans(o c_s^i) e T_gc
    for each orientation angle θ_j in n_os do
        e T_gc^{i,j} = e T_gc^i Rot(y, θ_j)
        e T_e^{i,j} = e T_gc^{i,j} g T_gc^{-1} e T_g^{-1}
        e C_g^{i,j} = e T_e^{i,j} e T_g g C_g
        if (EvalSE(t E_to, t E_tp, e T_t^{-1} e C_g^{i,j}, 10) < 1) then
            break
        end if
        if (EvalSE(g E_go, g E_gp, g T_c^{i,j c} C_o, 10) < 1) then
            break
        end if
        d_{i,j} = DistanceEU(g F_gc^{i,j} − g F_o^{i,j})
        g S_{i,j}^f = ∅, g C̄_o^{i,j} = ∅
        for each finger f in n_f do
            for each workspace sphere g s^f in g S^f do
                if EvalSphere(g s_o^f, g s_r^f, g C_o^{i,j}) < 1 then
                    g S_{i,j}^f = g S_{i,j}^f + g s^f, g C̄_o^{i,j} = g C̄_o^{i,j} + g c_o^{i,j}
                end if
            end for
        end for
        if g S_{i,j}^1 ≠ ∅ and ... and g S_{i,j}^{n_f} ≠ ∅ then
            if d_{i,j} < d* then
                g S_{i,j}^{f(*)} = g S_{i,j}^f, g C̄_o^{i,j(*)} = g C̄_o^{i,j}, e T_{e*} = e T_e^{i,j}
            end if
        end if
    end for
end for
return e T_{e*}
```

A. Gripper Pose Candidates

First, the origin of the gripper workspace centroid (GWC) frame $\mathcal{F}_{gc}$ is initialized with a gripper orientation perpendicular to the major axis of the object (represented by ${}^{e}\mathbf{T}_o$). This is done in the GWC frame for convenience, and the axis of rotation is gripper frame dependant, the rotation axes supplied in algorithm 1 corresponds to the Allegro right hand. Then for each point ${}^{o}c_s$ in the object bounding box sampling cloud ${}^{o}\mathcal{C}_s$, we first translate the GWC origin to the sample point ${}^{o}c_s^i$ with $i \in \{1 \dots n_s\}$, where $Trans()$ operation in Algorithm 1 denotes a $3D$ translation. This is done in the pre-rotated GWC frame, hence the pre-multiplication. Afterwards, we iterate through sampled orientation angle values $\theta_j = \theta_{init} + j * \theta_r/n_{os}$ about the object's major axis, with θ_{init}, θ_r, n_{os}, $j \in \{1 \dots n_{os}\}$ denoting the initial orientation angle, the angle range, number of orientation samples, and index respectively. Now that we have got the first orientation iteration at the first object sampled point, we transform it back to the effector frame to obtain the transformation candidate ${}^{e}\mathbf{T}_e^{i,j}$, applying which to the gripper point cloud, we obtain the first gripper pose candidate ${}^{e}\mathcal{C}_g^{i,j}$. Different gripper pose candidates are shown in Fig. 4 (a),(b),(c) for the Allegro right hand, and in Fig. 4 (d),(e),(f) for the Franka panda gripper. These poses correspond to the same object sampling point (per gripper), depicted by blue dot on the object (visible in Fig. 4 (a),(b),(d),(e) only), for different orientation samples.

B. Gripper Pose Evaluation

Firstly, the gripper pose candidate is evaluated against collision with the table, this is done by transforming the candidate into the table frame: ${}^{t}\mathcal{C}_g^{i,j} = {}^{e}\mathbf{T}_t^{-1}{}^{e}\mathcal{C}_g^{i,j}$ and substituting by each point of the obtained cloud in the table special ellipsoid (parameterized by ${}^{t}\mathcal{E}_{to}, {}^{t}\mathcal{E}_{tp}$) using (2). If (2) returns a value < 1 at any point of the gripper candidate cloud, this means collision with table is detected and this iteration terminates. Fig. 4 (a) and (d) depict gripper poses that collide with the table for Allego hand and Franka gripper respectively. Similarly the pose candidate is evaluated against collision with the object, this time by transforming the object ${}^{c}\mathcal{C}_o$ into the gripper candidate frame using the transform $({}^{e}\mathbf{T}_e^{i,j}{}^{e}\mathbf{T}_g)^{-1}{}^{e}\mathbf{T}_c$ in which the gripper "shape approximating" special ellipsoids are defined. Again, when using (2) returns a value < 1 at any point of the object "candidate" cloud, a collision with object is detected and the iteration terminates. Note that for object collision evaluation, we compute (2) iteratively for each element in the set of gripper SEs whose dimension is n_e. However, we chose not to include it in a separate for loop in Algorithm 1 to lighten the notation. Fig. 4 (b) and (e) depict gripper pose candidates that collide with the object for Allego hand and Franka gripper respectively.

If the gripper pose candidate does not collide with either the table or the object, we first compute the Euclidean distance $d_{i,j}$ for the current pose candidate between the GWC ${}^{g}\mathcal{F}_{gc}^{i,j}$ and the object ${}^{g}\mathcal{F}_o^{i,j}$ frames origin. Then we evaluate the transformed object point cloud ${}^{g}\mathcal{C}_o^{i,j}$ against each finger's set of workspace spheres ${}^{g}\mathcal{S}^f$ using (3), to construct a new subset ${}^{g}\mathcal{S}_{i,j}^f \subset {}^{g}\mathcal{S}^f$ of workspace spheres per finger

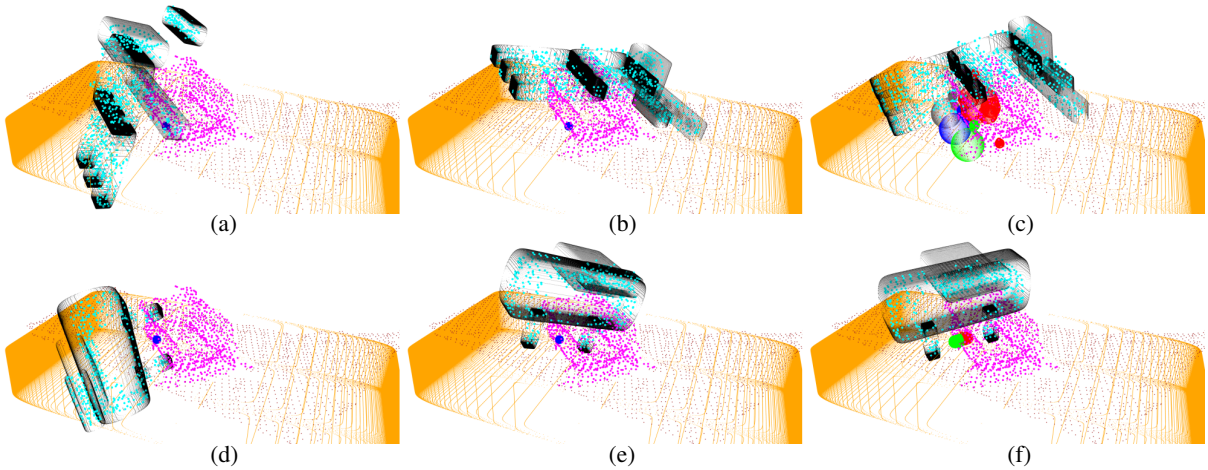


Fig. 4: Gripper pose candidates, with gripper point cloud and special ellipsoids in cyan and black respectively. Object point cloud in magenta, table special ellipsoid in orange. Finger workspace points in red (Allegro's thumb and Franka's right finger), green (Allegro's index and Franka's left finger), blue (Allegro's middle), and grey (Allegro's pinky).

that has at least one object cloud point inside as well as registering these object points in ${}^{g}\overline{\mathcal{C}}_{o}^{i,j} \subset {}^{g}\mathcal{C}_{o}^{i,j}$. In the final step, each newly constructed workspace spheres set ${}^{g}\mathcal{S}_{i,j}^{f}$ is checked not to be empty. In other words, this means that each gripper finger at the current pose candidate has a contact solution on the object. The algorithm then evaluates the best gripper pose based on its closeness to the object centroid by comparing $d_{i,j}$ and d^{*}, the latter being the Euclidean distance of the best gripper pose initialized to a large value at the beginning of the algorithm. Similarly, ${}^{g}\mathcal{S}_{i,j}^{f(*)}$, and ${}^{g}\overline{\mathcal{C}}_{o}^{i,j(*)}$ denote the best gripper workspace sphere sets and the corresponding object cloud points, initialized to null sets $\emptyset$ at the algorithm start. Using this approach, the best gripper pose together with the workspace spheres subsets are shown in Fig. 4 (c), and (f) for the Allegro right hand, and the Franka gripper respectively.

V. Experiments

In this section, the experimental results of the proposed grasping algorithm are presented, using the parameters provided in Table I. Two sets of experiments have been performed, one per gripper type. Each gripper was mounted to the Franka Emika arm (7 DoF), controlled in real-time with Franka control interface. The communication between the robot controller, the realsense camera, and the grippers is done through ROS. Motion planning is achieved using MoveIt! [27] based on the pose targets generated by our algorithm.

TABLE I: Grasping algorithm parameters

Parameter	Allegro Hand	Franka Gripper
n_e (special ellipsoids)	5	5
n_s (samples)	147	147
n_f (fingers)	4	2
n_{sp} (workspace spheres)	77	10
n_g (points)	870	499
n_o (points)	$300 \sim 800$	$300 \sim 800$
n_{os} (orientation samples)	4	4

A. Insights

Experiments feature 13 objects which were grasped with both the Allegro right hand, and the Franka 2 finger gripper, the objects we selected such that they are within the grasping volume dimensions of each gripper while maintaining a size/shape/texture variation. In sequence, the point cloud of the object is constructed from 3 view points using the Intel RealSense-D435 depth camera [28], the grasping algorithm computes a grasping pose based on the generated point cloud, the arm then moves to this pose, at which point the gripper performs the grasping action. The grasping action used in both grippers is a simple position control to a closed fingers configuration. Finally, the arm moves upward for 20 cm. An object is deemed grasped if it remains in static condition inside the gripper for more than 10 seconds. A sample of the grasped objects is shown in Fig. 5 by the Allegro right hand (upper) and the Franka gripper (lower), whereas Table II provides both the execution time (ET) and the success rate (SR) of each experiment.

The execution time reporte in Table II is measured only for the algorithm computation time in addition to point cloud registration and segmentation, since the point cloud acquiring process from several poses can be achieved more efficiently from multiple cameras mounted on the robot cell frame,

TABLE II: Grasping metrics per gripper for different objects

Object	Allegro Hand		Franka Gripper	
	ET	SR	ER	SR
Storage bin	$1.77s$	80%	$0.39s$	100%
Mug	$1.07s$	90%	$0.23s$	80%
Thermos	$1.61s$	40%	$0.33s$	0%
Realsense box	$1.39s$	100%	$0.41s$	90%
Cookies package	$1.24s$	80%	$0.62s$	40%
Plant pot	$1.43s$	100%	$0.36s$	80%
Plastic cup	$0.31s$	40%	$0.31s$	0%
Tooth paste	$1.64s$	60%	$0.24s$	100%
Toilet paper roll	$1.09s$	100%	$0.26s$	100%
Doll	$1.18s$	40%	$0.37s$	40%
Dish brush	$0.99s$	20%	$0.29s$	80%
Banana	$1.53s$	20%	$0.27s$	100%
Sprayer	$0.58s$	80%	$0.25s$	40%
Average	$1.22s$	65%	$0.34s$	65%

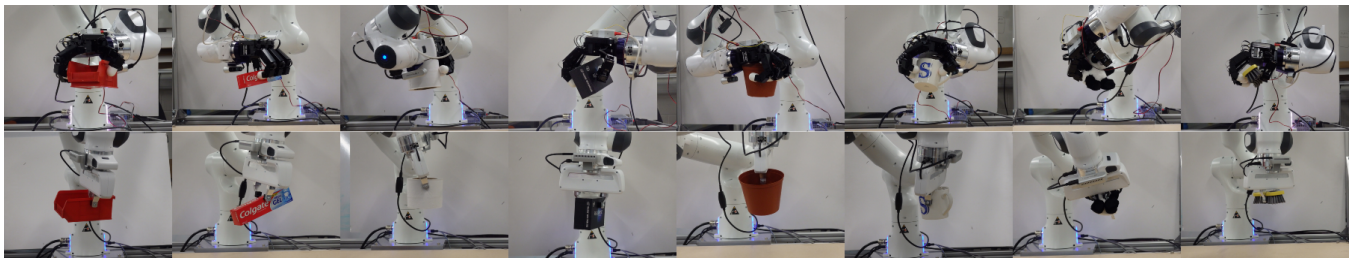


Fig. 5: Grasping several objects by both grippers, the Allegro right hand (upper) and the Franka panda gripper (lower). Objects from left to right: storage bin, tooth paste, toilet paper roll, realsense box, plant pot, mug, soft doll, and dish brush respectively.

a couple of Kinect2 depth sensors in opposite side facing configuration will be used in the future. Hence, we get a point cloud well representing the object in few milliseconds. Each object is grasped by each gripper for a total of 10 trials, out of which the success rate percentage is computed.

B. Discussion

In Table II, despite the relatively low average success rate per gripper, we can see that some objects can be grasped with high success rate, emphasizing the fact that some grippers are more suitable for grasping certain objects, mainly due to size constraints in the case of Allegro hand. Where the objects "banana", "Dish brush", have too low height to be grasped without hitting the table. The same applies to the Franka gripper, as the objects "Thermos", "Cookies package" were too big for the maximum grasping range to accommodate for, given the usual depth position error. On the other hand, the algorithm computation time is highly efficient with an average of 0.34, 1.22 seconds for the Franka gripper and Allegro hand respectively. The latter is due to the large number of workspace spheres per finger (77 in Table I) when compared to 10 spheres for 2 fingers in parallel jaw grippers. This can be reduced by optimizing the generated spheres filling up the finger workspace for multi-DoF hands, which is expected to have high impact on the computation time. In order to enhance the success rate, the authors propose exploring more gripper orientations in case of the parallel jaw grippers and to add an optimization step to select grasping points for high DoF hands. The former points motivates our future work in addition to applying the algorithm on other gripper types.

VI. Conclusion

In this work a novel grasping algorithm based on finger workspace spheres has been introduced. It is versatile and can applied to any type of gripper, here applied to a complex hand with 16 DoF as well as a simple jaw gripper with 2 DoF. The average computation time is very low ranging from 0.3 to 1.7 seconds depending on the complexity of the gripper. Successful experiments have been conducted to validate the proposed approach.

ACKNOWLEDGMENTS

This work is funded by EPSRC under grant agreement EP/R02572X/1, UK National Center for Nuclear Robotics initiative (NCNR).

REFERENCES

[1] J. K. Salisbury, J. K., "Kinematic and force analysis of articulated mechanical hands," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 105, no. 1, pp. 35–41, 1983.

[2] V.-D. Nguyen, "Constructing force closure grasps," *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.

[3] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, May 1992, pp. 2290–2295.

[4] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.

[5] S. Jain and B. Argall, "Grasp detection for assistive robotic manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2015–2021.

[6] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *2003 IEEE International Conference on Robotics and Automation (ICRA)*, Sep. 2003, pp. 1824–1829 vol.2.

[7] V. Lippiello, F. Ruggiero, B. Siciliano, and L. Villani, "Visual grasp planning for unknown objects using a multifingered robotic hand," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 1050–1059, June 2013.

[8] J. Baumgartl and D. Henrich, "Fast vision-based grasp and delivery planning for unknown objects," in *ROBOTIK 2012; 7th German Conference on Robotics*, May 2012, pp. 1–5.

[9] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis ;a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, April 2014.

[10] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4304–4311.

[11] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[12] Q. Lei, J. Meijer, and M. Wisse, "Fast c-shape grasping for unknown objects," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2017, pp. 509–516.

[13] B. S. Zapata-Impata, "Using geometry to detect grasping points on 3d unknown point cloud," in *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. SciTePress, 2017, pp. 154–161.

[14] K. Huebner and D. Kragic, "Selection of robot pre-grasps using box-based shape approximation," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2008, pp. 1765–1770.

[15] M. Przybylski, T. Asfour, and R. Dillmann, "Unions of balls for shape approximation in robot grasping," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2010, pp. 1592–1599.

[16] A. H. Quispe, B. Milville, M. A. Gutirrez, C. Erdogan, M. Stilman, H. Christensen, and H. B. Amor, "Exploiting symmetries and extrusions for grasping household objects," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3702–3708.

[17] Q. Lei and M. Wisse, "Fast grasping of unknown objects using force balance optimization," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2014, pp. 2454–2460.

[18] Q. Lei, J. Meijer, and M. Wisse, "A survey of unknown object grasping and our fast grasping algorithm-c shape grasping," in *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, April 2017, pp. 150–157.

[19] T. Suzuki and T. Oka, "Grasping of unknown objects on a planar surface using a single depth image," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2016, pp. 572–577.

[20] Y. Lin, S. Wei, and L. Fu, "Grasping unknown objects using depth gradient feature with eye-in-hand rgb-d sensor," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2014, pp. 1258–1263.

[21] M. Adjigble, N. Marturi, V. Ortenzi, V. Rajasekaran, P. Corke, and R. Stolkin, "Model-free and learning-free grasping by local contact moment matching," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 2933–2940.

[22] C. Eppner and O. Brock, "Grasping unknown objects by exploiting shape adaptability and environmental constraints," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2013, pp. 4000–4006.

[23] FrankaEmika.com, "Franka-emika panda gripper," https://www.franka.de/panda/.

[24] WonikRobotics.com, "Allegro hand," http://www.simlab.co.kr/Allegro-Hand.htm.

[25] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[26] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[27] I. A. Sucan and S. Chitta, "Moveit!" [Online]Available:http://moveit.ros.org.

[28] Intel.com, "Intel realsense depth camera," https://click.intel.com/intelr-realsensetm-depth-camera-d435.html.