Fast grasping of unknown objects using cylinder searching on a single point cloud

Lei, Qujiang; Wisse, Martijn

DOI
10.1117/12.2268422

Publication date
2017

Document Version
Final published version

Published in
Ninth International Conference on Machine Vision

PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Fast grasping of unknown objects using cylinder searching on a single point cloud

Qujiang Lei, Martijn Wisse

SPIE.

Fast grasping of unknown objects using cylinder searching on a single point cloud

Qujiang Lei[a], Martijn Wisse[b]
TU Delft Robotics Institute, Delft University of Technology, The Netherlands
[a] q.lei@tudelft.nl, [b] m.wisse@tudelft.nl

ABSTRACT

Grasping of unknown objects with neither appearance data nor object models given in advance is very important for robots that work in an unfamiliar environment. The goal of this paper is to quickly synthesize an executable grasp for one unknown object by using cylinder searching on a single point cloud. Specifically, a 3D camera is first used to obtain a partial point cloud of the target unknown object. An original method is then employed to do post treatment on the partial point cloud to minimize the uncertainty which may lead to grasp failure. In order to accelerate the grasp searching, surface normal of the target object is then used to constrain the synthetization of the cylinder grasp candidates. Operability analysis is then used to select out all executable grasp candidates followed by force balance optimization to choose the most reliable grasp as the final grasp execution. In order to verify the effectiveness of our algorithm, Simulations on a Universal Robot arm UR5 and an under-actuated Lacquey Fetch gripper are used to examine the performance of this algorithm, and successful results are obtained.

Keywords: unknown object grasping, cylinder searching, single point cloud, 3D vision, robot.

1. INTRODUCTION

Grasping of unknown objects with neither appearance data nor object models given in advance is very important for robots that work in an unfamiliar environment. The motivation of this paper is to quickly synthesize an executable grasp on the unknown object for the under-actuated grippers shown in Fig.1 (c) and (d). Fig.1 (a) shows a spray bottle which works as an example of unknown objects to explain our proposed grasping algorithm. Fig.1 (b) shows a robot arm equipped with a 3D camera and an under-actuated gripper, in which the 3D camera is used to obtain the point cloud of the target object and the under-actuated gripper is used to do final grasp execution.

(a) (b) (c) (d)

Fig.1. The motivation of this paper: (a) shows an example of an unknown object, (b) shows a robot arm equipped with a 3D camera and an under-actuated gripper, (c) and (d) show two types of under-actuated grippers respectively. The motivation of this paper is to quickly synthesize an executable grasp on the unknown object for the under-actuated grippers shown in (c) and (d).

Vast research has been conducted on the problem of unknown object grasping and many achievements have been obtained in the previous years. However, unknown object grasping is still a challenging task that has not yet been solved in a general manner. [1] gives a profound survey about unknown object grasping. The existing unknown object grasping algorithms can be divided into two main categories, namely, using partial model and using full model.

The first method is building a full 3D model using many images or point clouds of the target object. In [2], the full 3D model is fit and split into many minimum volume bounding boxes and a grasp is found on these bounding boxes. In [3], two flat, parallel surfaces are found on the 3D model to realize the grasping task with a gripper. In [4], the center of mass and axes of inertia of the target object are calculated from the 3D model, and then a grasp on the center or along the axes is found. [5] uses a genetic algorithm to search for grasping points on a 3D model of the target object. [6] uses a

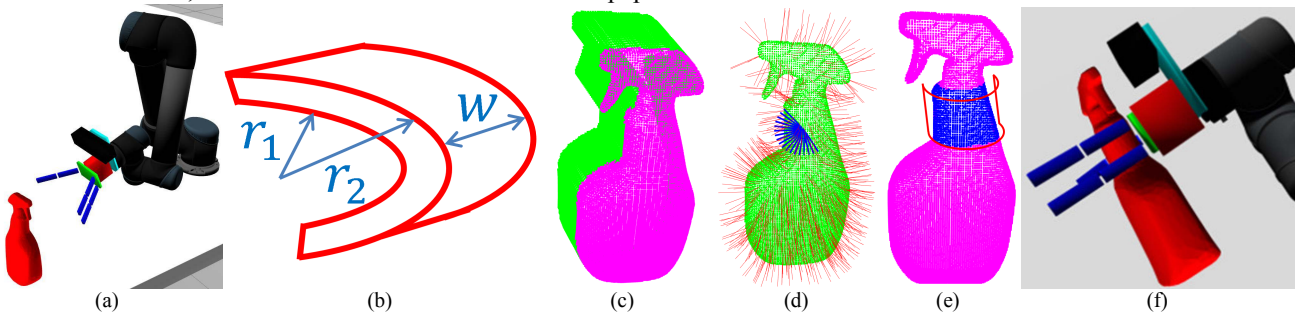
Ninth International Conference on Machine Vision (ICMV 2016), edited by Antanas Verikas, Petia Radeva, Dmitry P. Nikolaev, Wei Zhang, Jianhong Zhou, Proc. of SPIE Vol. 10341, 1034108

cost function to analyze the 3D model to obtain grasping points. In [7], the 3D model is simplified into some shape primitives (boxes or cylinders). Then grasping points which are assigned offline to these shape primitives are selected for the corresponding shape. [8] establishes a benchmark for the object grasping community by building a grasp simulator which is called GraspIt. After GraspIt, OpenGRASP [9] is invented on the base of the OpenRAVE [10], which made a progress comparing with GraspIt. OpenGRASP uses the normal of the object as the approaching vector of the robot hand, which can greatly reduce the number of grasp candidates. However, [11] states that depending on the choice of the parameters, the time of using OpenGRASP to simulate all the corresponding grasp candidates for a common object can vary from a few minutes to more than an hour. [11] uses [12] to do sampling of approaching vectors to further reduce the grasp candidates. However, it still needs about one minute to find a good grasp for the unknown object, which is pretty time-consuming. Therefore, we can get a short summary about using full 3D model to compute grasp, that is, using full 3D model is usually time consuming. However, [9] and [11] illuminate us to use down-sampled normals to accelerate grasping searching. So, in this paper, we will focus on how to quickly synthesize a suitable grasp if one approaching normal is chosen.

The second method is using partial information of an object to achieve fast grasping. [13] uses partial object geometry to achieve a semantic grasp. This algorithm needs predefined example grasps and cannot deal with the grasping task of symmetric objects since multiple views of a symmetric object could have the same depth images. [14] proposes a data-driven grasp planner that requires partial sensor data. Matching and alignment methods were used for grasping after obtaining the Columbia Grasp Database. [15] uses local descriptors from several images to construct the 3D model of an object. Object registration was conducted by using a set of training images. [16] installs a 2D range sensor on the robot at an inclined angle to acquire partial shape information of the unknown objects. Two straight lines are extracted directly from this partial shape information as the two grasp sides for a parallel jaw gripper. [17] uses binocular vision to recover the partial 3D structure of unknown objects. Then process the incomplete 3D point clouds searching for good grasp candidate for a three finger robot hand according to a function that accounts for both the feasibility and the stability. In a short summary, using partial information can surely decrease the computation time of grasp searching, however, using partial information inevitably introduces some uncertainty which may lead to grasp failure. Therefore, in this paper, we will use partial information of the unknown object to compute grasp to accelerate grasp searching, in the same time, we need to focus on how to deal with the uncertainty which we may encounter if we use partial information of the target object.

In our previous work [18, 19, 20, 21], we used features (principal axis, concavity and boundary) of the object to find suitable grasp. In this paper, we will start from the shape of the under-actuated gripper we use to solve the problem of unknown object grasping. Our method is to simplify the gripper as a two-layer cylinder (shown as Fig.1 (d) and Fig.2 (b)) with radius r_1 and r_2 respectively, then, the algorithm will do cylinder searching on the single point cloud of the target object to quickly synthesize an executable grasp. Fig.2 is the outline of our proposed method. Specifically, Fig.2 (a) shows a simulation setup consisting of a robot arm equipped with a 3D camera, an under-actuated gripper and a spray bottle working as an example of an unknown object, Fig.2 (b) shows the inspiration of this paper, the gripper in Fig.1 (d) can be described as a two-layer cylinder with radius r_1 and r_2, Fig.2 (c) shows our approach to deal with the unseen part of the target object, Fig.2 (d) demonstrates our method to reduce the grasp candidates to accelerate the speed of grasp searching, Fig.2 (e) shows an executable grasp (the blue part) found by our algorithm, Fig.2 (f) shows the grasp execution for the spray bottle. Details about our algorithm will be explained in section 2.

This paper is organized as follows: section 2 contains a detailed explanation of our algorithm, sections 3 shows the simulation results, and section 4 is the conclusion of this paper.



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

Fig.2. The outline of our grasping algorithm.

2. DETAILED ALGORITHM

This section contains detailed explanation of our grasping algorithm. Specifically, 2.1 shows the problem formulation; 2.2 explains how to obtain the point cloud of the target object; 2.3 illustrates how to do grasp searching; 2.4 demonstrates how to choose the best grasp by using force balance optimization.

2.1 Problem formulation

As mentioned above, we simplify the under-actuated gripper we use as a two-layer cylinder. Then, the algorithm will do cylinder searching on the single point cloud of the target object to quickly synthesize an executable grasp. In order to get the parametric equations for an arbitrary cylinder in 3D space, we need to first know how to obtain the parametric equations for an arbitrary circle on an arbitrary plane. C (x_0, y_0, z_0) is used to stand for the center of the arbitrary circle and r is its radius. If the arbitrary plane is II and its unit normal vector is N=($\cos a$, $\cos\beta$, $\cos\gamma$), among which a , β , γ are the direction angles of the unit normal, then the arbitrary plane II can be obtained by transforming the XOY plane through the following transformation: rotating around the X axis by a , rotating around the Y axis by β , then moving along the vector N to (x_0, y_0, z_0) . The whole transformation can be summarized as equation (1).

$$T=\begin{bmatrix}1&0&0&0\\0&\cos a&\sin a&0\\0&-\sin a&\cos a&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}\cos\beta&0&\sin\beta&0\\0&1&0&0\\\sin\beta&0&\cos\beta&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\x_0&y_0&z_0&1\end{bmatrix}=\begin{bmatrix}\cos\beta&0&-\sin\beta&0\\\sin a\sin\beta&\cos a&\sin a\cos\beta&0\\\cos a\sin\beta&-\sin a&\cos a\cos\beta&0\\x_0&y_0&z_0&1\end{bmatrix} \tag{1}$$

If ($x(t)$, $y(t)$, $z(t)$) is used to sand for an arbitrary point on the arbitrary circle, the parametric equations of the circle can be obtained using equation (2) and (3), in which $0\le t\le 2\pi$.

$$\begin{pmatrix}x(t)\\y(t)\\z(t)\\1\end{pmatrix}=(r\cos t\ \ r\sin t\ \ 0\ \ 1)\begin{bmatrix}\cos\beta&0&-\sin\beta&0\\\sin a\sin\beta&\cos a&\sin a\cos\beta&0\\\cos a\sin\beta&-\sin a&\cos a\cos\beta&0\\x_0&y_0&z_0&1\end{bmatrix} \tag{2}$$

$$\begin{cases}x(t)=x_0+r\cos t\cos\beta+r\sin t\sin a\sin\beta\\y(t)=y_0+r\sin t\cos a\\z(t)=z_0+r\sin t\sin a\cos\beta-r\cos t\sin\beta\end{cases} \tag{3}$$

If the cylinder alignment is $r(t)=\{x(t),y(t),z(t)\}$, and the unit normal vector of the cylinder is $N=(\cos a,\cos\beta,\cos\gamma)$, the parametric equations for an arbitrary cylinder in 3D space can be obtained using equation (4), in which $0\le t\le 2\pi$ and $0\le s\le w$, w is the width of the griper (w is shown as Fig.2 (b)).

$$\begin{cases}x(s,t)=x_0+r\cos t\cos\beta+r\sin t\sin a\sin\beta+s\cos a\\y(s,t)=y_0+r\sin t\cos a+s\cos\beta\\z(s,t)=z_0+r\sin t\sin a\cos\beta-r\cos t\sin\beta+s\cos\gamma\end{cases} \tag{4}$$

Then the problem can be formulated as follows: finding a , β (a and β make up the general rotation of the transformation matrix in equation (1)), so that the difference between $P_c\ (x(t),y(t),z(t))$ (point on the cylinder) and $P_o\ (x(o),y(o),z(o))$ (point on the object) can be minimized, which can be summarized as equation (5).

$$f_{\min}=\sum\left(\begin{pmatrix}x(s,t,a,\beta)\\y(s,t,a,\beta)\\z(s,t,a,\beta)\end{pmatrix}-\begin{pmatrix}x(o)\\y(o)\\z(o)\end{pmatrix}\right) \tag{5}$$

In equation (5), $0\le t\le 2\pi$, $0\le s\le w$, a and β stand for the cylinder orientations in 3D space. From the above analysis, the cylinder searching problem can be simplified as the searching of a and β .

2.2 Obtaining the point cloud of the target object

The raw point cloud from the 3D sensor contains the environment (for example the table plane). In order to quickly isolate the point cloud of the target object, down-sampling and distance filtering are first applied on the raw point cloud obtained by the 3D camera to reduce the computation time and remove the points out of the reach of the robot arm. Then Random Sample Consensus (RANSAC) method is used to remove the table plane, resulting in the isolated point cloud of the target object (shown as the purple points in Fig.2 (c)).

2.3 Grasp searching

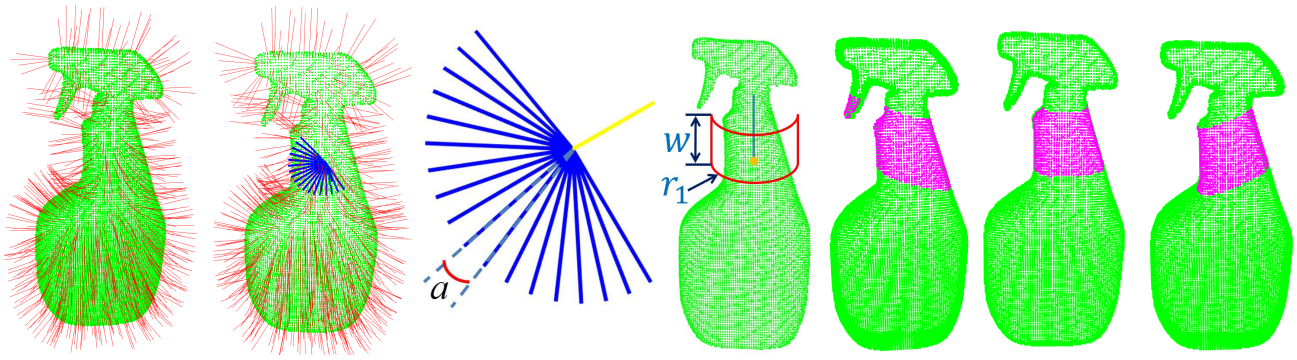
According to section 2.1, if the center point and the axis direction of the cylinder are figured out, the cylinder can be worked out. Therefore, the problem in section 2.1 can be divided into two parts, one is to find the orientation of the cylinder, and the other is to find the center point of the cylinder. That is to find the normal vector N($\cos\alpha$, $\cos\beta$, $\cos\gamma$) and the center point P_{center} (x_0, y_0, z_0).

2.3.1 Determination of cylinder axis

As mentioned above, the normal of the target object is used to simplify the grasp searching from SE(3) to SE(2). If a random normal N_r is chosen, then cylinder axis can only rotate around N_r, then we can search incrementally around N_r with an incremental angle a (shown in Fig.3 (c)). Fig.3 presents a clear and understandable example. Specifically, the red lines in Fig.3 (a) represent the normals of the target object. The blue lines in Fig.3 (b) work as the cylinder axis incrementally allocated around a rand normal (the yellow line in Fig.3 (c) stands for the random normal). The blue line in Fig.3 (d) stands for one cylinder axis. The red frame in Fig.3 (d) stands for the corresponding cylinder. Fig.3 (e), (f) and (g) are three example cylinder orientations, and the purple areas stand for the corresponding point cloud covered by the three cylinders.

Point cloud from the 3D camera is located in the camera coordinate system, which should be transformed to the local coordinate system to do following analysis. Fig.4 shows the point cloud transformation from the camera coordinate system (CCS) to the local coordinate system (LCS). Fig.4 (a) shows the point cloud in the camera coordinate system. Fig.4 (b) shows the relation between the CCS and the LCS. Ω_c and Ω_l respectively stand for the point cloud in CCS and the point cloud in LCS. Fig.4 (c) shows an example point cloud in LCS. $\vec{n}$, $\vec{p}$ and $\vec{q}$ in Fig.4 (b) mean the unit direction vector of the coordinate axis of LCS. (x_0, y_0, z_0) stands for the translation between LCS and CCS. So the transformation matrix (T_{l_c}) between LCS and CCS can be formulated as equation (6). The transformation from Ω_c to Ω_l can be achieved using the inverse matrix of T_{l_c} ($\Omega_l = \Omega_c T_{l\_c}^{-1}$).

$$T_{l\_c} = \begin{pmatrix} n_x & p_x & q_x & x_0 \\ n_y & p_y & q_y & y_0 \\ n_z & p_z & q_z & z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$



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

Fig.3. Normal of the target object is used to simplify the grasp searching and explaination of how to orientate the cylinder.

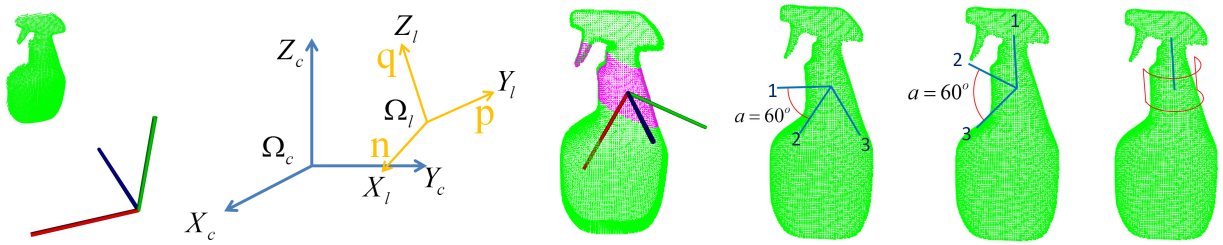


Fig.4. Point cloud transformation from the camera coordinate system to the local coordinate system ((a) to (c)), and how to increase the possiblity of finding suitable grasp ((d) to (f)).

During the generation of the cylinder axis, a parameter a is introduced. If a is a big angle, for example, a is 60^o in Fig.4 (d) and (e), we may get two totally different allocations of cylinder axis. In Fig.4 (d), the three cylinder axis will lead to no suitable grasps found, because the gripper will collide with the object. However, the cylinder axis 1 in Fig.4 (e) corresponds to a very good grasp candidate (shown in Fig.4 (f)). What result in the difference? The answer is the location of the first cylinder axis. Therefore, we need to focus on how to generate the first cylinder axis. Here, we propose to use the principal axis of the local point cloud to work as the first cylinder axis. If the local point cloud which use point on the rand normal as its center and r_1 as its radius is abstracted, principal component analysis (PCA) is performed to approximate the principal axis of the local point cloud. PCA is a statistical technique for analyzing correlation between observed data. Let $X = (\chi_1, \chi_2, \cdots, \chi_n)$ be the object point set, where χ_i is a point in R^3. The object position point $P_{centroid}$ is calculated as $p_{centriod} = \frac{1}{n}\sum_{i=1}^{n} \chi_i$. From X, the variance covariance matrix is calculated by equation (7).

$$s = \begin{pmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{xz}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{yz}^2 \\ \sigma_{zx}^2 & \sigma_{zy}^2 & \sigma_{zz}^2 \end{pmatrix} \tag{7}$$

Then the eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$, and the corresponding eigenvectors μ_1, μ_2, μ_3 of the variance covariance matrix s are obtained. The eigenvector μ_1 corresponds to the largest eigenvalue λ_1, which approximates the direction of the principal axis and it is used as the direction of the first cylinder axis.

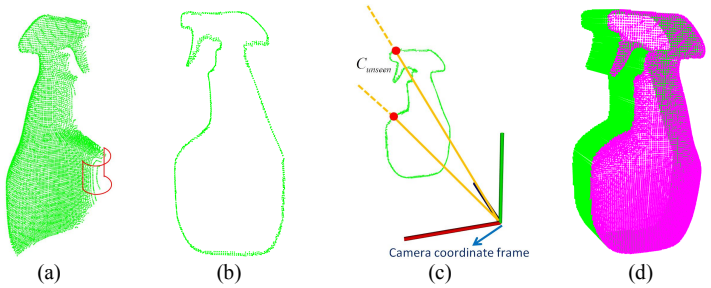
2.3.2 How to deal with the unseen part

In this section, we will explain our method to deal with the problem of uncertainty (unseen part) produced by using partial information of the target object. Fig.5 (a) shows an example that the gripper will collide with the target object if we do not consider the unseen part, which may result in grasp failure. In this paper, we propose to employ the boundary of the object to eliminate the uncertainty led by the unseen part. Specifically, the point cloud in the camera coordinate system is used to work out the boundary points Ω_b (shown as Fig.5 (b)). Fig.5 (c) shows our idea. In detail, the two red points belong to Ω_b, the two orange lines are obtained by connecting the origin point of camera coordinate system and the two red points. The two orange dashed lines are obtained by extending the two orange lines. This method will go through all the points on the boundary, and we can obtain a point cloud shown as Fig.5 (d). Then, the configuration space (C space) of the target object (C_{obj}) is divided into two parts. C'_{obj} and C_{unseen} are used to describe the configuration spaces after the unseen part is generated ($C_{obj} = C'_{obj} + C_{unseen}$).

2.3.3 Determination of cylinder center point

After the orientation of cylinder axis is obtained, we need to determine the center point of the cylinder. As we mentioned before, the under-actuated gripper will approach the object along the normal direction. Then a question comes out, that is, where to stop?

Fig.6 is used to explain how to determine the center point of the cylinder. Fig.6 (a) is a possible grasp candidate, the green points stand for the points covered by cylinder. Fig.6 (b) is the abstracted point cloud, and the red arrow stands for the approaching direction. The two red points in Fig.6 (b) are two example cylinder center points. And the two blue circles are the corresponding cylinder. It is obvious to find that the two cylinder center points are not the best one. The

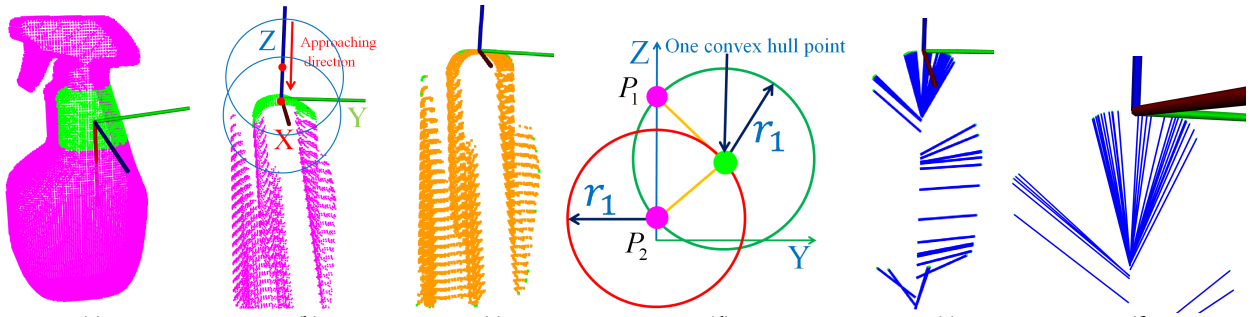


(a) (b) (c) (d)

Fig.5. How to deal with the unseen part of the target object to eliminate the grasp uncertainty.

cylinder center point can go down further. (c), (d), (e) and (f) are used to explain how to determine the cylinder center point. The abstracted point cloud in Fig.6 (b) is first projected to the YOZ plane to get the projected point cloud (orange points shown as Fig.6 (c)). Then the convex hull of the projected point cloud is extracted shown as the green points in Fig.6 (c). The green point in Fig.6 (d) stands for one convex hull point. If we draw a circle with r_1 as its radius (shown as the green circle), we can obtain two intersects with Z axis (shown as the two purple points P_1 and P_2). Then $Z = \min(Z_1, Z_2)$ will work as the cylinder center. Using this method goes through all the green points in Fig.6 (c), we can get all the center points $Z_c = (Z_{c1}, Z_{c2}, \cdots, Z_{cn})$ (shown as Fig.6 (e)). The maximal Z_c is used as the final cylinder center (shown as the equation (8)). The maximal cylinder center point means the earliest contact point with the object when the gripper approaches the object.

$$Z_{c\_\max} = \max(Z_{c1}, Z_{c2}, \cdots, Z_{cn}) \tag{8}$$

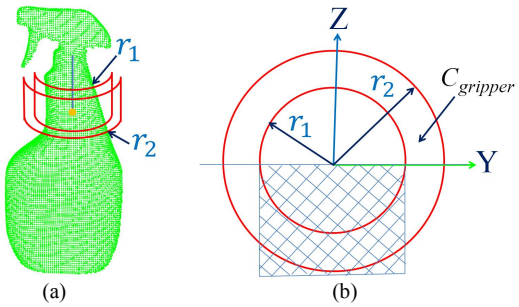


(a) (b) (c) (d) (e) (f)

Fig.6. How to determine the cylinder center point.

2.3.4 Grasp judgement

After the cylinder orientation and the cylinder center point are determined, we will check whether the cylinder can envelop the object. Fig.7 (a) shows an example of cylinder. In the YOZ plane of the local cylinder coordinate system, the two cylinders of the gripper are shown as two red circles in Fig.7 (b). Z axis is used as the approaching direction, so the shadow areas will not affect grasp execution. If Ω_p is used to stand for the projected point cloud, Ω_S is used to stand for the points in the shadow area, and Ω_{p-s} ($\Omega_{p-s} = \Omega_p - \Omega_s$) is used to stand for the projected point cloud without the shadow part, the gripper configuration space in Fig.7 (b) can be described as $C_{gripper} = \{q \mid q \in C_{r1-r2} \wedge q \notin C_s\}$.



(a) (b)

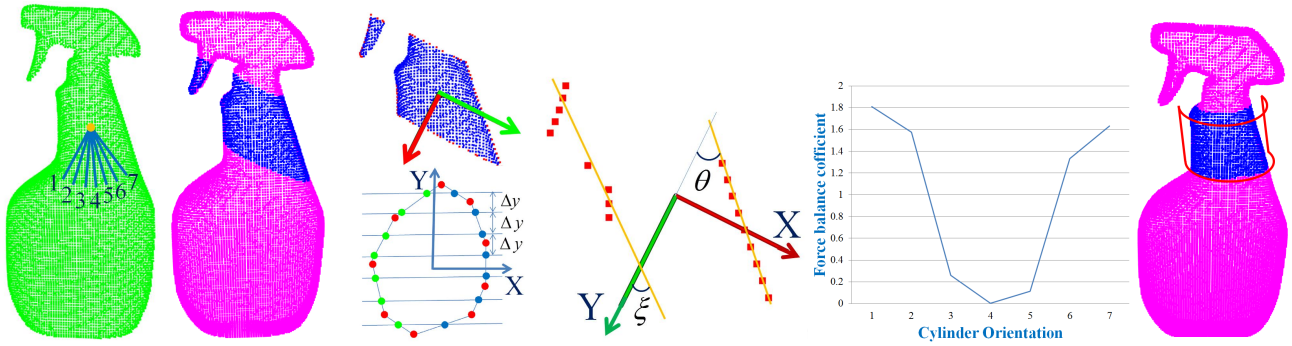
Fig.7. Collision check with the gripper.

Then the problem is simplified to find whether there are points in gripper configuration space $C_{gripper}$. If there are no points in $C_{gripper}$, it means that this cylinder can envelop the object and this grasp candidate will be reserved. If there are points in $C_{gripper}$, it means that this cylinder will collide with the object and this grasp should be removed. In the end of this section, we can get a vector ($G=(g_1,g_2...g_n)$) which is used to store all grasp candidate without colliding with the object.

2.4 Force balance optimization

All grasp candidates in the vector ($G=(g_1,g_2...g_n)$) can be executed without collision with the object. If 1, 2, 3, 4, 5, 6 and 7 in Fig.8 (a) stand for the grasp in the vector $G=(g_1,g_2...g_7)$, then how to choose the best one as the final grasp.

Here, we propose to use force balance optimization to select out the best grasp. Usually, the existing papers will employ the physic property to do force balance computation, for example, the friction coefficient. But in our case, we cannot know the physic property, because the objects for this paper are unknown. We propose to use the local geometry shape to do force balance computation. The blue points in Fig.8 (b) stand for the grasp candidate 1 (g_1). It is projected to the XOY plane to get the projected point cloud shown as the upper in Fig.8 (c). Then we can extract the concave hull of the projected point cloud shown as the red points in the upper in Fig.8 (c). The red points in the lower in Fig.8 (c) stand for an example concave hull. Straight lines along the Y axis are allocated with an incremental distance Δy. The most left and most right intersection point between each line and the concave hull are extracted. The most left points are shown as green ones and the most right points are shown as blue ones. Fig.8 (d) shows the most left points and most right points of the grasp candidate in Fig.8 (b). If the points are $((x_1,y_1),(x_2,y_2)...(x_n,y_n))$, a straight line ($y=kx+b$) can be fit out by using equation (9). The two orange lines in Fig.8 (d) stand for the two fit lines for left side and the right side. θ and ξ are used to stand for the angel between the fit lines and the Y axis. σ is used to stand for the sum of θ and ξ, i.e. $\sigma=\theta+\xi$. σ_s is used to stand for the absolute value of σ. The bigger σ_s is, the less force balance is. The vector $\sigma_s=(\sigma_{s1},\sigma_{s2}...\sigma_{s7})$ is used to store all the angles for the grasp vector $G=(g_1,g_2...g_7)$. Fig.8 (e) is a line graph for the vector σ_s, the grasp with the smallest σ_s is chosen as the final grasp. Fig.8 (f) is the best grasp returned, which corresponds to the 4th grasp candidate in Fig.8 (a) and (e).



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

Fig.8. Force balance optimization.

$$\begin{cases} k=\dfrac{\sum_{i=1}^{n}x_iy_i-\frac{1}{n}\sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i}{\sum_{i=1}^{n}x_i^{\,2}-\frac{1}{n}\sum_{i=1}^{n}x_i\sum_{i=1}^{n}x_i} \\ b=\dfrac{1}{n}\sum_{i=1}^{n}y_i-k\dfrac{1}{n}\sum_{i=1}^{n}x_i \end{cases} \tag{9}$$

3. SIMULATION

In order to verify our grasping algorithm, several objects in different geometry shapes are chosen to do simulations. All the tested objects can be seen in the first column in Fig.9. The second column in Fig.9 shows the generation of the normal lines and the cylinder axis. The third column shows the unseen part the target object. The fourth column illustrates the result of force balance computation. And the last column shows the best grasp returned. Table I shows the number of the points of every tested object and the computation time. We can find that the algorithm can quickly work out the best grasp within 2 seconds for one chosen normal line.

Table I. Simulation results of the grasp computation

Unknown Objects	Spray bottle	Electrical drill	Table tennis racket	gun	Water bottle
points	8297	7400	6274	4183	6874
Time(s)	2.068	1.515	1.353	1.194	1.498

4. CONCLUSION

In this paper, a novel algorithm of unknown object grasping is presented for under-actuated grippers. The grippers are simplified as a cylinder, which is used to do cylinder search on the point cloud of the target object to find suitable grasp candidates for the robot. In order to accelerate the computation speed, this algorithm only uses a signal point cloud as input. The number of grasp candidates can be greatly reduced by using the normal line of the target object to guide the generation of the grasp candidates. Meanwhile, we propose an original method to deal with the unseen part of the object to enhance the grasp security. In order to verify the effectiveness of our algorithm, several objects commonly used by other grasping algorithms with different geometric shapes were used to do simulations, and successful results are obtained.

ACKNOWLEDGEMENT

The work leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 609206.

REFERENCES

[1] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven graspsynthesis—a survey," IEEE Transactions on Robotics, VOL. 30, NO. 2, pp. 289–309, 2014.
[2] Hubner K, Kragic D, "Selection of robot pre-grasps using box-based shape approximation," in IROS, pp. 1765–1770, 2008.
[3] Bone GM, Lambert A, Edwards M, "Automated modeling and robotic grasping of unknown three dimensional objects," in ICRA, pp. 292–298, 2008.
[4] E. Lopez-Damian, D. Sidobre, and R. Alami, "A grasp planner based on inertial properties," in ICRA, pp. 754–759, 2005.
[5] H.-K. Lee, M.-H. Kim, and S.-R. Lee, "3D optimal determination of grasping points with whole geometrical modeling for unknown objects," Sensors and Actuators, vol. 107, pp. 146–151, 2003.
[6] K. Yamazaki, M. Tomono, and T. Tsubouchi, "Picking up an Unknown Object through Autonomous Modeling and Grasp Planning by a Mobile Manipulator," Field and Service Robotics, Springer Tracts in Advanced Robotics, vol. 42, pp. 563–571, 2008.
[7] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in ICRA, pp. 1824–1829, 2003.
[8] Andrew Miller and Peter K. Allen, "Graspit!: A Versatile Simulator for Robotic Grasping," IEEE Robotics and Automation Magazine, vol. 11, no. 4, pp.110–122, 2004.
[9] B. Le´on, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales,T. Asfour, S. Moisio, J. Bohg, J. Kuffner, and R. Dillmann, "OpenGRASP: A toolkit for robot grasping simulation," in SIMPAR '10:Proceedings of the 2st International Conference on Simulation, Modeling, and Programming for Autonomous Robots, pp. 109–120, 2010.
[10] R. Diankov and J. Kuffner, "Openrave: A planning architecture forautonomous robotics," Robotics Institute, Pittsburgh, PA, Tech. Rep.CMU-RI-TR-08-34, 2008.
[11] J. Bohg, M. Johnson-Roberson, B. Leon, J. Felip, X. Gratal, N. Bergstrom, D. Kragic, and A. Morales, "Mind the gap–robotic grasping under incomplete observation," in ICRA, pp. 686–693, 2011.
[12] M. Richtsfeld and M. Vincze, "Grasping of Unknown Objects from a Table Top," in *ECCV Workshop on 'Vision in Action: Efficientstrategies for cognitive agents in complex environments'*, Marseille,France, 2008.

[13] Hao Dang and Peter K. Allen, “Semantic Grasping: Planning Robotic rasps Functionally Suitable for An Object Manipulation Task,” in IROS, pp. 1311-1317, 2012.
[14] Goldfeder C, Ciocarlie M, Peretzman J, Dang H, Allen PK, “Data-driven grasping with partial sensor data,” in IROS, pp. 1278–1283, 2009.
[15] Collet A, Berenson D, Srinivasa SS, Ferguson D, “Object recognition and full pose registration from a single image for robotic manipulation,” in ICRA, pp. 3534–3541, 2009.
[16] Zhaojia Liu, Lounell B. Gueta, and Jun Ota, “Feature Extraction from Partial Shape Information for Fast Grasping of Unknown Objects,” in ROBIO, pp. 1332–1337, 2011.
[17] I. Gori, U. Pattacini, V. Tikhanoff, and G. Metta, “Three-finger precision grasp on incomplete 3D point clouds,” in ICRA, pp. 5366–5373, 2014.
[18] Qujiang Lei, Martijn Wisse, “Fast grasping of unknown objects using force balance optimization,” in IROS, pp. 2454–2460, 2014.
[19] Qujiang Lei, Martijn Wisse, “Unknown object grasping using force balance exploration on a partial point cloud,” in AIM, pp. 7–14, 2015
[20] Qujiang Lei, Martijn Wisse, “Unknown object grasping by using concavity,” accepted by 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2016.
[21] Qujiang Lei, Martijn Wisse, “Object grasping by combining caging and force closure,” accepted by 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2016.

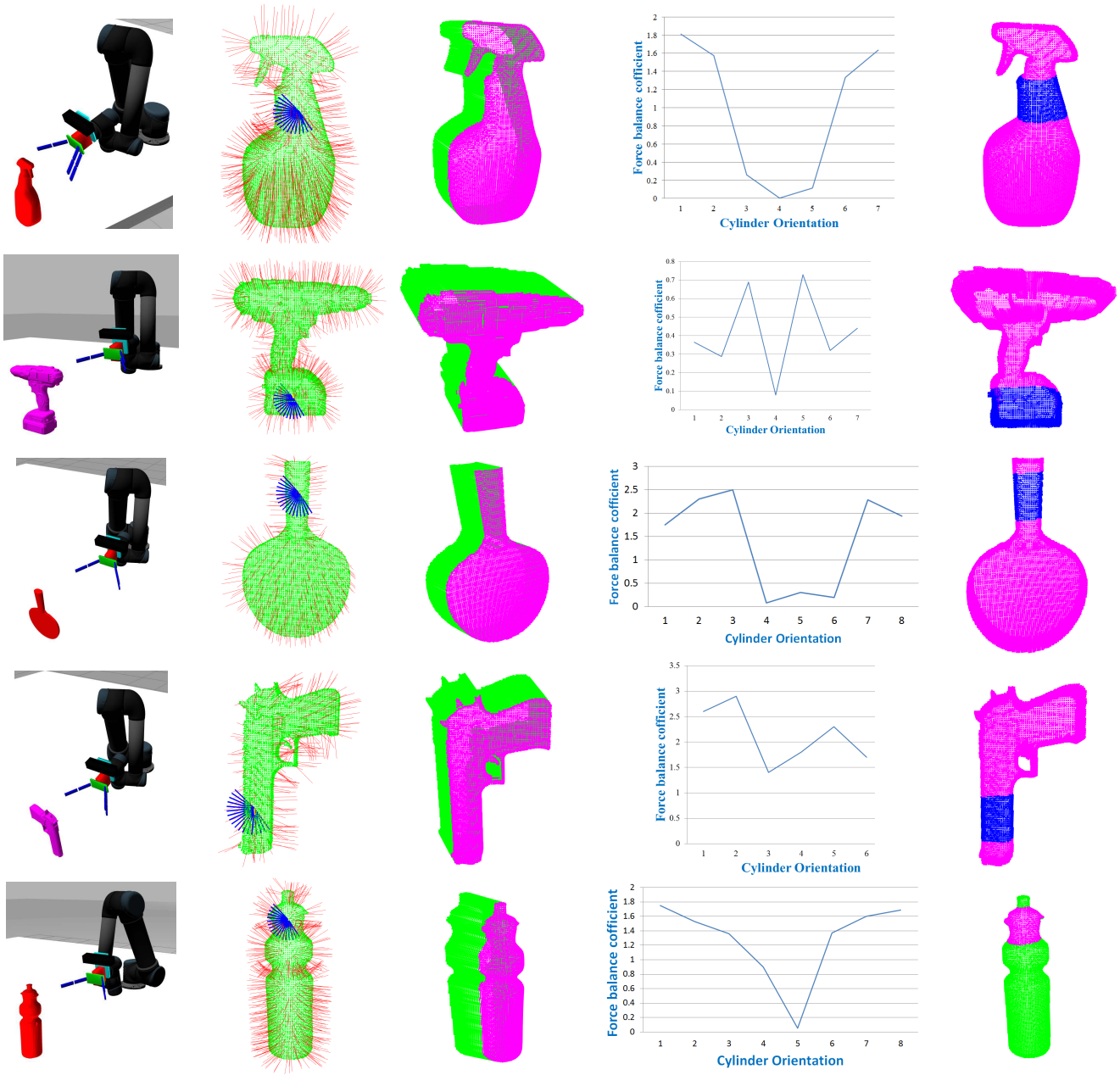


Fig.9. Simulation results.