



Delft University of Technology

## Fast grasping of unknown objects using principal component analysis

Lei, Qujiang; Chen, Guangming; Wisse, Martijn

**DOI**

[10.1063/1.4991996](https://doi.org/10.1063/1.4991996)

**Publication date**

2017

**Document Version**

Final published version

**Published in**

AIP Advances

**Citation (APA)**

Lei, Q., Chen, G., & Wisse, M. (2017). Fast grasping of unknown objects using principal component analysis. *AIP Advances*, 7(9), Article 095126. <https://doi.org/10.1063/1.4991996>

**Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

# Fast grasping of unknown objects using principal component analysis

Qujiang Lei,<sup>a</sup> Guangming Chen, and Martijn Wisse

*Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Delft, Netherlands*

(Received 25 June 2017; accepted 18 September 2017; published online 29 September 2017)

Fast grasping of unknown objects has crucial impact on the efficiency of robot manipulation especially subjected to unfamiliar environments. In order to accelerate grasping speed of unknown objects, principal component analysis is utilized to direct the grasping process. In particular, a single-view partial point cloud is constructed and grasp candidates are allocated along the principal axis. Force balance optimization is employed to analyze possible graspable areas. The obtained graspable area with the minimal resultant force is the best zone for the final grasping execution. It is shown that an unknown object can be more quickly grasped provided that the component analysis principle axis is determined using single-view partial point cloud. To cope with the grasp uncertainty, robot motion is assisted to obtain a new viewpoint. Virtual exploration and experimental tests are carried out to verify this fast grasping algorithm. Both simulation and experimental tests demonstrated excellent performances based on the results of grasping a series of unknown objects. To minimize the grasping uncertainty, the merits of the robot hardware with two 3D cameras can be utilized to suffice the partial point cloud. As a result of utilizing the robot hardware, the grasping reliance is highly enhanced. Therefore, this research demonstrates practical significance for increasing grasping speed and thus increasing robot efficiency under unpredictable environments. © 2017 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/1.4991996>

## I. INTRODUCTION

Unknown object means an item that has neither geometric model nor appearance information. Grasping unknown objects is highly challenging for the robots working at unfamiliar environments.<sup>1</sup> With the increasing demand of various robots that are being used in contemporary society, increasing grasping speed becomes one of the primary tasks for improving the efficiency of robots manipulation.<sup>2</sup>

A vast amount of research has been conducted on grasping unknown objects over the past few decades, and many achievements have been attained. To grasp an unknown object, geometric properties (i.e. symmetries,<sup>3</sup> surface,<sup>4</sup> edges,<sup>5–10</sup> boundary,<sup>11–15</sup> silhouette,<sup>16</sup> saliency<sup>17</sup>) are generally used to construct contours of the target object. For instance, Maldonado et al.,<sup>12</sup> ten Pas and Platt<sup>13</sup> fitted the shape of the gripper on the boundary of partial point cloud of the target unknown object. To obtain geometric contours of the unknown object, two methods are commonly used. One is to use tactile sensors to detect the geometric properties of unknown target object.<sup>18–27</sup> The other is to use a camera to move around to explore the unseen part.<sup>16,28,29</sup> Both methods have high grasp security but are very time expensive. The reason for the first method is that it requires long time to carry out sufficient contacts with the object. For the second method, much time is used because of the movement of the camera.

---

<sup>a</sup>Contact author: E-mail: [q.lei@tudelft.nl](mailto:q.lei@tudelft.nl)

To save time, several fast grasping approaches have been proposed, which can be found in Refs. 6,8,13,15, and 28. Among these approaches, Johannes Baumgartl<sup>8</sup> uses RGB images as input, which can quickly provide 2D geometric information of an unknown object. However, using this approach cannot always promote successful grasping because an unknown object can have no parallel edges. By contrast, the other researchers<sup>6,13,15,28</sup> employ a partial point cloud, which can formulate more realistic geometric model. Because the geometry contour is approximated based on partial point cloud, it can significantly reduce the computing load and thus accelerating grasping speed. Nevertheless, ignoring other information, such as occlusions, may introduce grasp uncertainty and result in grasp failure. To deal with the uncertainty with the usage of partial point cloud, a new method for reducing geometric information for grasping unknown objects can be explored.

The goal of this paper is to reduce grasping time for unknown objects whilst the grasping security is maintained. In this paper, we propose a novel approach to guide the grasping procedures of unknown objects based on the principal component analysis. Based on this, a single-view point cloud is used to reduce the data for formulating geometric contour to save the computational time. The feature of our grasping approach is to allocate grasp candidates along the principal axis such that the possibility of useless grasp candidates can be greatly decreased. This algorithm is shown to be successful on the base of both simulation and experimental tests. By taking the advantage of robot hardware, the grasping uncertainty is minimized. Therefore, this research demonstrates practical significance for increasing grasping speed.

This paper is organized in this way: section II introduces our fast grasping algorithm; section III shows the simulation results; section IV gives the experiment validation; section V outlooks an approach on enhancing the grasp security using two 3D cameras. Finally, the conclusion of this research is provided in section VI.

## II. A FAST GRASPING APPROACH

This section presents a detailed explanation of our fast grasping approach for unknown objects. This approach adopts a grasping algorithm which utilizes a single-view partial point cloud. Furthermore, the solutions for tackling exceptional cases of grasping failure by applying this algorithm are elaborated.

### A. Algorithm

Because the configuration of the robot hand follows a Special Euclidean group  $SE(3)$  in practice, it implies many possibilities when locating a robot hand in three-dimensional (3D) space. In our approach, the principal axis of the target unknown object is used to find out proper positions for executing a successful grasping action. Figure 1 outlines our fast grasping algorithm, in which it shows a single-view partial point cloud of the target object is used as input. For general case of grasping unknown objects, seven steps are required, the details of which are described in section II B. For the exceptional case to achieve a successful grasping, the solution is illustrated in section II C.

### B. Grasping unknown object based on the single-view partial point cloud

Figure 2 presents the procedure to grasp the target unknown object based on the single-view partial point cloud. Figure 2 (a) shows a simulation setup in which a spray bottle is used as the target unknown object. An eye-in-hand system is composed of a 3D camera sensor and a UR5 robot. The 3D camera sensor is used to acquire the raw point cloud for the given environment. In order to accelerate computing speed, distance filtering is initially applied on the raw point cloud to remove those points that are out of the reach of the robot arm, as shown in Figure 2 (b). Figure 2 (c) shows the transformation of the partial point cloud to the world frame. Figure 2 (d) illustrates the transformation of the partial point cloud to the object frame. Figure 2 (e) gives the projected point cloud in the object frame. Figure 2 (f) presents the concave hull contour of the projected point cloud. Figure 2 (g) depicts all the crossing points. Figure 2 (h) shows possible grasp zone within grasping range of the gripper. Figure 2 (i) points the method to obtain the best grasp on the graspable zone. Finally, Figure 2 (j)

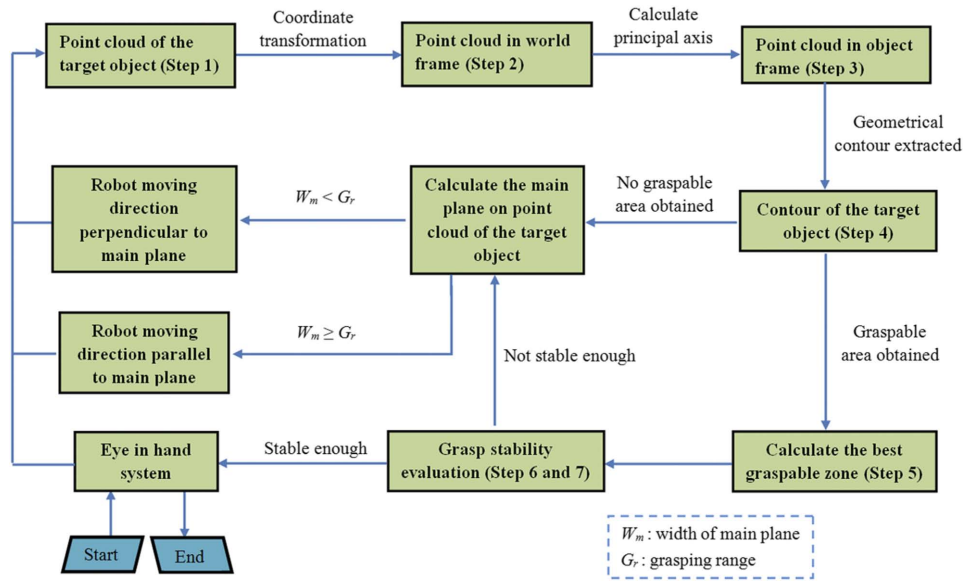


FIG. 1. Overview of our fast grasping algorithm.

provides an example of grasp execution. The detailed seven steps for conducting grasping an unknown object are presented as follows.

### Step 1: Obtaining the single-view partial point cloud of the target unknown object

Figure 2 (a) shows a simulation setup in which a spray bottle is used as the target unknown object. An eye-in-hand system is composed of a 3D camera sensor and a UR5 robot. The 3D camera sensor is used to acquire the raw point cloud for the given environment. In order to accelerate computing speed, distance filtering is initially applied on the raw point cloud to remove those points that are out of the reach of the robot arm, as shown in Figure 2 (b). Then down sampling is used to reduce the density of the points. After that, Random Sample Consensus (RANSAC) method is employed to remove the table plane. The determination of principal axis is given as follows.

After the single-view partial point cloud is obtained by a 3D camera, Principal Component Analysis (PCA) is performed to approximate the centroid and the principal axis of the object. PCA

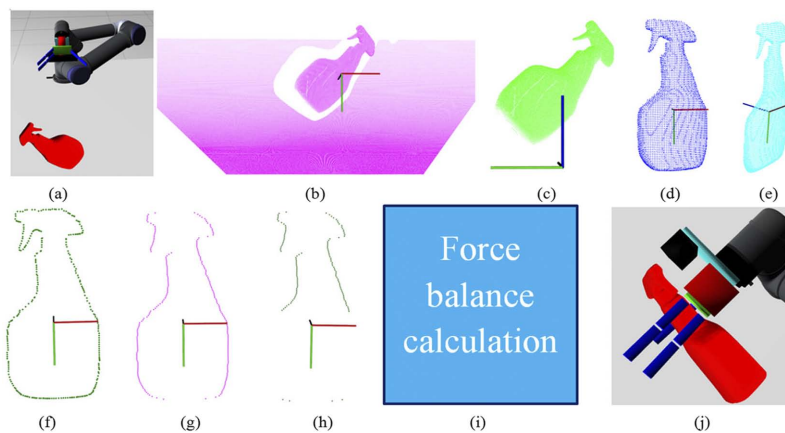


FIG. 2. The procedure to process the single-view partial point cloud. (a) simulation setup; (b) the filtered distance and down-sampled point cloud. (c) transformation of the partial point cloud to the world frame; (d) transformation of the partial point cloud to the object frame; (e) projected point cloud in the object frame; (f) concave hull contour of the projected point cloud. (g) all the crossing points; (h) possible grasp zone within grasping range of the gripper. (i) the method to obtain the best grasp on the graspable zone. (j) an example of successful grasp execution.

is a statistical technique for analyzing correlation between observed data. Let  $X = (\chi_1, \chi_2, \dots, \chi_n)$  be the object point set, where  $\chi_i$  is a point in the 3D space  $R^3$ . The centroid of the point set  $P_{centroid}$  is calculated by the following equation:

$$P_{centroid} = \frac{1}{n} \sum_{i=1}^n \chi_i \quad (1)$$

Giving the values of a point cloud  $X = (\chi_1, \chi_2, \dots, \chi_n)$ , the covariance matrix  $s$  can be calculated by using equation (2):

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

in which the nine elements are the values of covariance for the 3D coordinates. The eigenvalues  $\lambda_1 > \lambda_2 > \lambda_3$  ( $\lambda_i \in R$  and  $i=1, 2, 3$ ), and the corresponding eigenvectors  $\mu_1, \mu_2, \mu_3$  ( $\mu_i \in R^3$  and  $i=1, 2, 3$ ) can be obtained. The eigenvector  $\mu_1$  corresponds to the largest eigenvalue  $\lambda_1$ , which approximates the direction of the principal axis. Using the centroid ( $P_{centroid}$ ) of the single-view partial point cloud and the direction of the principal axis, the principal axis can be obtained.

### Step 2: Transforming single-view partial point cloud from camera sensor frame to world frame

The obtained single-view partial point cloud in step 1 is retained in the camera frame. To carry out analysis using coordinate system, the single-view partial point cloud must be transformed into the world frame. Feedbacks from the joint encoders of the robot arm are used to construct the transformation matrix from the end effector of the robot to the base link of the robot. The transformed single-view partial point cloud in the world frame can be seen in Figure 2(c). The transformed single-view partial point cloud in the object frame is shown in Figure 2 (d).

### Step 3: Constructing object frame

Figure 3 (a) displays three coordinate systems from the eye-in-hand system and the target unknown object, namely, the world frame ( $X_{world} Y_{world} Z_{world}$ ), the 3D camera frame ( $X_{3Dsensor} Y_{3Dsensor} Z_{3Dsensor}$ ) and the object frame ( $X_{obj} Y_{obj} Z_{obj}$ ). The principle axis and the mass center of the partial point cloud are used to build the object frame. As can be seen from Figure 3 (b), the mass center of the target object is used as the origin point of the object frame.

The principal axis is used as the Y axis of the object frame. The X axis and the Z axis of the object frame can be determined by applying equation (3).

$$\begin{cases} \overrightarrow{O_c X} = \overrightarrow{O_c P_s} \times \overrightarrow{O_c P_p} \\ \overrightarrow{O_c Z} = \overrightarrow{O_c Y} \times \overrightarrow{O_c X} \end{cases} \quad (3)$$

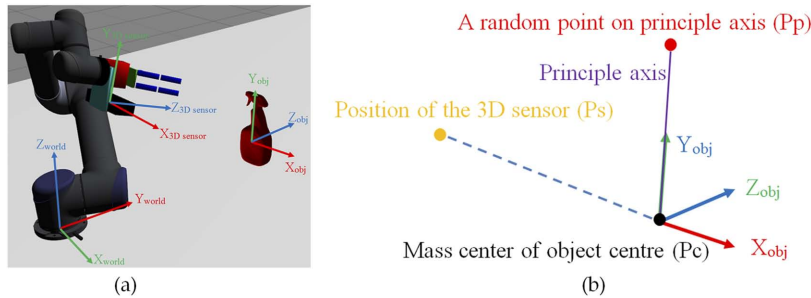


FIG. 3. Establishment of the coordinate systems. (a) world frame, the 3D sensor frame and the object frame. (b) illustration of building the object frame of the target unknown object.

in which  $P_p$  and  $P_s$  respectively stand for a random point on the principal axis and the position of the 3D camera sensor.

#### Step 4: Acquiring concave hull contour of the single-view partial point cloud

After we obtained the partial point cloud in the object frame, the contour of the partial point cloud will be abstracted. Specifically, the partial point cloud is firstly projected to the XOY ( $X_{obj}O_{obj}Y_{obj}$ ) plane of the object frame. Figure 2 (e) shows the projected point cloud. There are two methods to obtain the contour of the projected point cloud, namely, the concave hull contour and the convex hull contour. For this scenario, concave hull contour can better represent the geometric shape of the target unknown object. Concave hull contour of the target object is extracted as shown in Figure 2(f).

#### Step 5: Calculating graspable zones

To figure out all graspable area within grasping range on the contour of the target object, grasp candidates are allocated along the principal axis (the Y axis of the object frame). The minimum Y value ( $y_{min}$ ) of all the points on the concave hull contour is firstly decided. Subsequently, a straight line parallel to the X axis is used during the searching process from the top to the bottom. In this manner, the most left and most right crossing points between the straight line and the concave hull contour can be determined. An appropriate step ( $\Delta y$  shown in Figure 4) is added to  $y_{min}$  such that the searching process can apply to the whole concave hull contour from the top to the bottom.

The straight line parallel to the X axis can be obtained according to equation (4). All crossing points can construct a point cloud as shown in Figure 2(g). For each straight line, when the distance between the most left point and the most right is smaller than the grasping range of the robot hand, the most left point and the most right point will be saved to construct a point cloud, which is shown in Figure 2(h). The graspable zone of the target unknown object is repented by the point cloud in Figure 2(h).

$$y = y_{min} + n\Delta y \quad (4)$$

#### Step 6: Optimizing total force on the XOY plane of the object frame

In the step 5, the graspable zone is extracted out by considering the grasp range of the robot hand. To obtain the best grasp on the graspable zone, Figure 5 illustrates the evaluation process to allocate grasp candidates along the principal axis from the top to the bottom. As shown in Figure 5(a), the green, blue and red rectangles stand for three example grasp candidates. In order to achieve a stable final grasp, force balance computation is carried out for every grasp candidate. The blue points in Figure 5(b) stand for a grasp candidate. Points on the two grasp sides are used to fit two straight lines, and the angle between the two straight lines is used to evaluate the stability of this grasp. The bigger the angle is, the less stable the grasp is.

The straight line can be expressed as  $y = kx + b$ , in which the coefficients  $k$  and  $b$  can be determined using equation (5). The two red lines in Figure 5(c) stand for the two fitting lines. Figure 6 demonstrates the results of force balance computation of the graspable zone. We can observe that force balance reaches the minimum value when the number comes to search index 33, which corresponds to the best force balance.

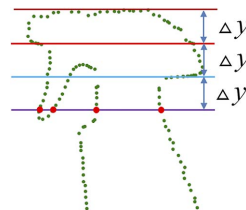


FIG. 4. The method to determine the graspable zone of the target unknown object.

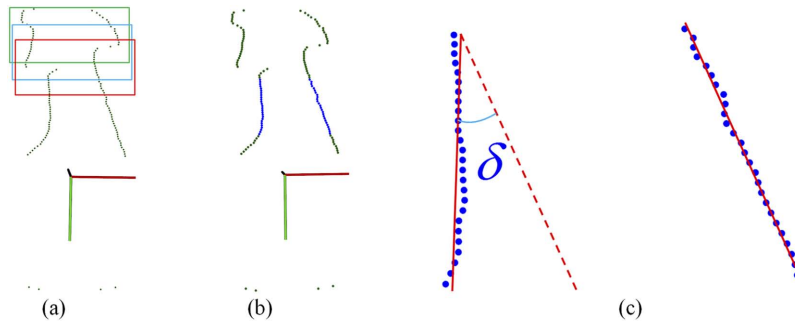


FIG. 5. Evaluation of the grasping candidates within grasp range (a) green, blue and red rectangles of grasp candidates; (b) blue points of a grasping candidate (c) two fitting lines.

$$\left\{ \begin{array}{l} k = \frac{\sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sum_{i=1}^n x_i^2 - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n x_i} \\ b = \frac{1}{n} \sum_{i=1}^n y_i - k \frac{1}{n} \sum_{i=1}^n x_i \end{array} \right. \quad (5)$$

### Step 7: Force balance evaluation on the XOZ plane of the object frame.

In the previous step, we explained the method to determine the best grasp using force balance computation on the XOY plane of the object frame. Next, the effect of force balance on the XOZ plane on the stability must be also studied. Figure 7(a) shows an initial configuration of the robot and the target unknown object. Figure 7(b) shows that the robot is approaching the grasp point. Figure 7(c) shows the force balance analysis of this grasp candidate on the XOZ plane.  $F_1$  and  $F_2$  respectively stand for the force that the gripper imposes on the target object.

The point cloud covered by the grasp candidate is extracted to be used for force balance analysis on the XOZ plane, which is illustrated in Figure 8. The green points in Figure 8(a) stand for the area covered by a grasp candidate. Figure 8(b) shows the point cloud covered by grasp candidate is extracted and the two grasp sides are shown as the red points in Figure 8(b). The average Z values of the left and the right grasp sides are worked out shown as the  $Z_1$  and  $Z_2$  in Figure 8 (c). The difference between  $Z_1$  and  $Z_2$  is used to evaluate the stability of the grasp candidate on the XOZ plane. A threshold ( $Z_{dif\ max}$ ) can be set by this system, thus, when  $|Z_1 - Z_2| < Z_{dif\ max}$ , the grasp candidate is saved, otherwise the grasp candidate is removed.

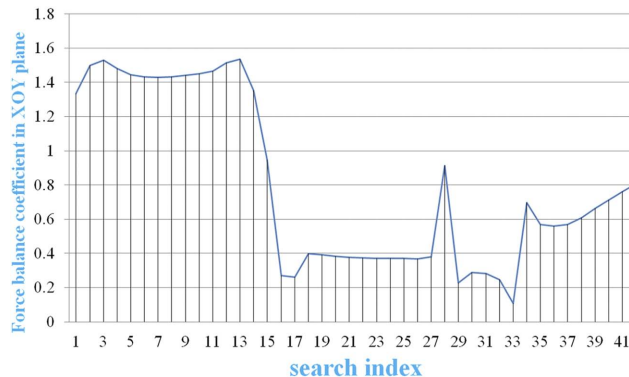


FIG. 6. The results of force balance computation on the XOY plane.



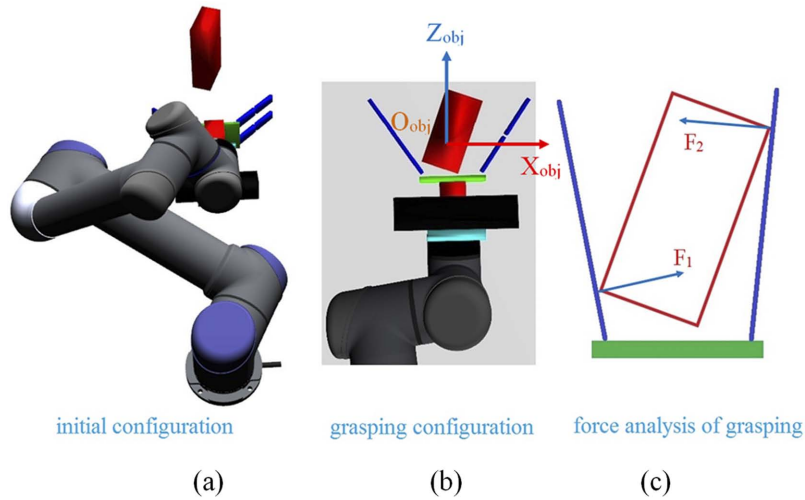


FIG. 7. Evaluation of the grasp candidates on the XOZ plane (a) initial configuration of the robot and the target unknown object. (b) the robot becomes contact with the grasp point. (c) the force balance analysis of this grasp candidate on the XOZ plane.

The above seven steps illustrate the fast grasping strategy of using principal component analysis based on the obtained single-view point cloud. However, there are four risks that can lead to grasping failures. Firstly, a grasping failure can occur when the principle axis cannot be obtained using single-view point. Figure 9 illustrates three successful grasps from the three perspectives of left, middle and right by rotating the robot arm. It also includes a grasping failure due to the fact that principle axis was not determined because of the point data loss.

Secondly, a graspable zone is not possible to be obtained when the gripper cannot cover the target object, which can be deduced in step 5. Thirdly, the angle ( $\delta$ ) corresponding to the force balance calculation can be too large (as shown in step 6), which infers that the object will be squeezed out when the robot tries to perform the grasping action. In addition, grasping failure can be triggered because of the resultant unbalanced force when the range between  $Z_1$  and  $Z_2$  for all grasp candidates are greater than the threshold (step 7).

Nevertheless, the grasping failure that is caused by the unbalanced force on the XOZ in step 7 can be resolved by using changing viewpoint. The solution for this exceptional case was already pointed out in Figure 1 of this algorithm. In the following subsection, the detailed explanation of the solution to the this exceptional case is provided.

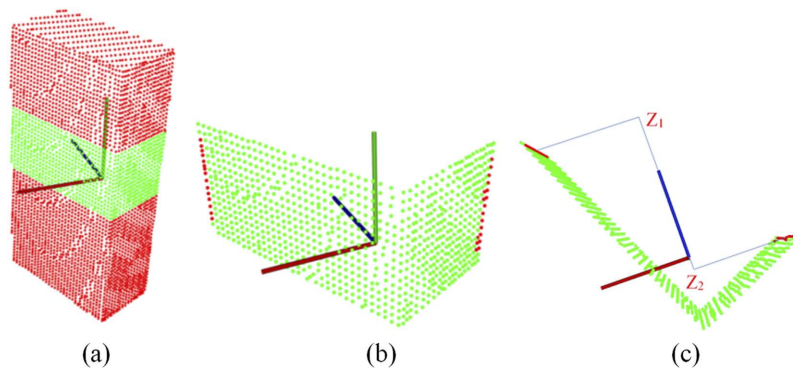


FIG. 8. Evaluation of grasping an object on the XOZ plane. (a) the green area of point cloud covered by a grasp candidate. (b) extracted grasping areas and the two grasp sides of the red points. (c) The average values for Z of the left and the right grasping sides.



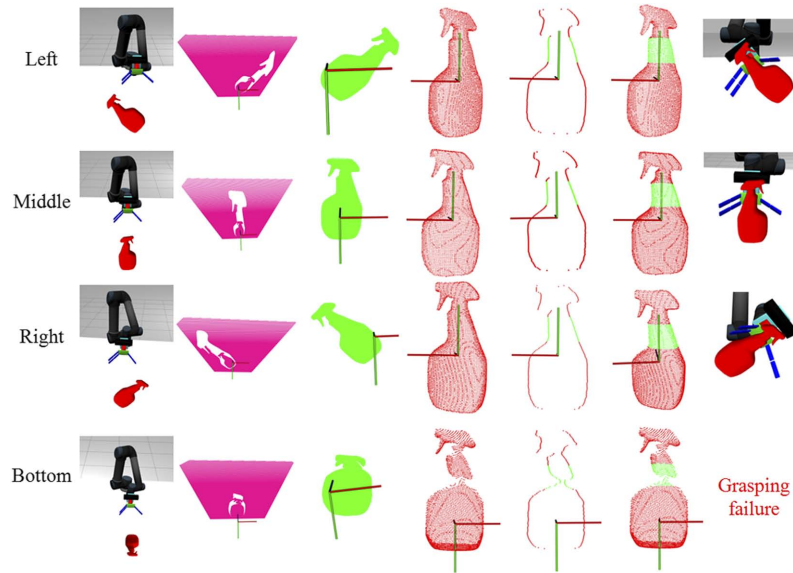


FIG. 9. Grasping the same object from four different perspectives (left, middle, right and bottom) using a single-view point cloud by following the seven steps.

### C. Solution for the exceptional case

To deal with the exceptional case that grasping positions cannot be achieved at one viewpoint, the main plane to guide the robot arm can be employed to activate another motion of the robot arm. In this way, a robot can move to another viewpoint to calculate to search for an executable grasp by following the above seven steps.

Figure 10 illustrates the activation of another motion of the robot arm, in which Figure 10 (a) and (d) provides two scenarios of two possible graspable areas obtained by the grasp algorithm. The green points in Figure 10 (b) and Figure 10 (e) respectively stand for the corresponding best grasping areas returned from the grasping algorithm at this viewpoint. The blue points in Figure 10 (c) and (f) respectively stand for the main plane of the grasp in Figure 10 (b) and (e). For the scenario as seen in Figure 10 (a), the shortest distance of movement of the robot arm is to move to the perpendicular direction of the main plane as shown in Figure 11 (a). For the other scenario shown in Figure 10 (d), the shortest distance of movement of the robot arm is to move to the tangent direction of the main plane as shown in Figure 11 (b).

Referring to Figure 1, when the width of the main plane ( $W_m$ ) is smaller than the graspable range of the gripper ( $G_r$ ), then the robot will move to the perpendicular of the main plane. This situation corresponds to for the Figure 11 (a) which is obtained from Figure 10 (a). The angle ( $\beta$ ) of the movement between the initial sensor point to the target sensor point should be  $\beta = \pi/2 - a$ . Figure 11(b) shows the other situation which is that the width ( $d$ ) of the main plane ( $W_m$ ) is bigger than the grasp range of the gripper ( $G_r$ ). The angle ( $\beta$ ) of the movement between the initial sensor point to the target

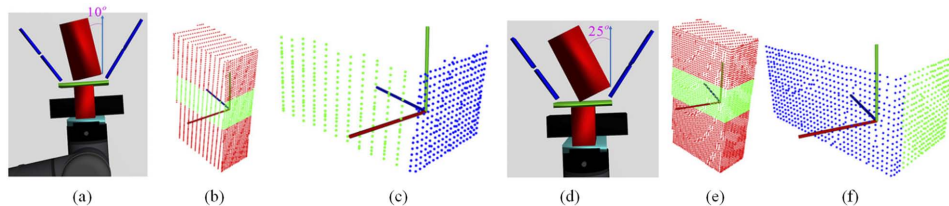


FIG. 10. The method to deal with the exceptions that no suitable grasp is found at a viewpoint plane. (a) and (d) two cases of best grasp found by the grasp algorithm; (b) and (e) the corresponding best grasping areas; (c) and (f) the main plane.

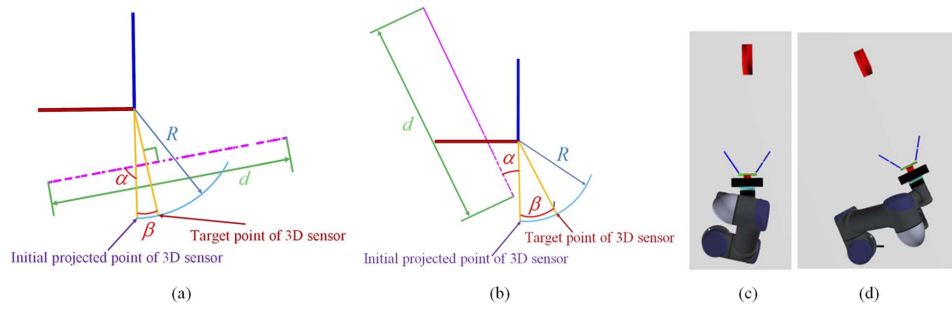


FIG. 11. Strategies of moving the robot arm when no suitable grasp is found at one viewpoint, in which (a) and (b) are the obtained projected main plane (two purple lines), (c) and (d) are illustrations of the robot arm approaching the target sensor point.

sensor point should be  $\beta = \alpha$ . The specific rotation can be worked out using equation (6). Figure 11 (c) and (d) shows the robot arm arrives at the target sensor point corresponding to Figure 10 (a) and (d). When the robot arm arrives at the target viewpoint, the steps from step 1 to step 7 are repeated to search a suitable grasping area for the target object.

$$\begin{bmatrix} P_{i,x} \\ P_{i,y} \\ P_{i,z} \end{bmatrix} = \begin{bmatrix} x^2(1-c) + c & xy(1-c) - zs & xz(1-c) + ys \\ yx(1-c) + zs & y^2(1-c) + c & yz(1-c) - xs \\ xz(1-c) - ys & yz(1-c) + xs & z^2(1-c) + c \end{bmatrix} \begin{bmatrix} P_i,x \\ P_i,y \\ P_i,z \end{bmatrix} \quad (6)$$

To sum up, the application of our algorithm of using principal component analysis has been elaborated based on the seven steps. To solve the grasping failure of an exceptional case, robot motion is used to obtain single-view point data from another perspective. It can be inferred that our novel algorithm of using single-view partial point cloud can save time compared to other algorithms that used partial point cloud data. To verify and validate the efficiency and applicability of our approach, simulation and experimental tests are presented in the subsequent two sections.

### III. SIMULATION TEST

In this section, our grasping algorithm is verified in a simulation environment. First, the structure of the simulation setup are illustrated. Next, the simulations for grasping unknown objects with different geometric shapes are performed using a single-view partial point cloud.

#### A. Structure of simulation setup

Figure 12 illustrates the simulation setup that consists of ROS, Gazebo and MoveIt!. ROS (an open source robot operating system) is widely used in the community of robotics due to the simply operations. Gazebo (an Open Dynamics Engine simulator) is the state of art simulator that offers the ability to efficiently and accurately simulate complex task for robots. MoveIt! (a cutting edge

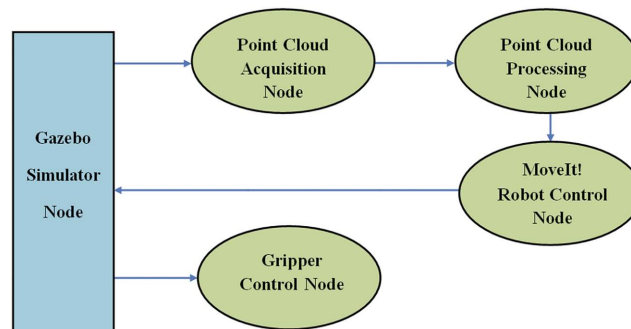


FIG. 12. Illustration of simulation setup consisting of ROS, Gazebo and MoveIt!.

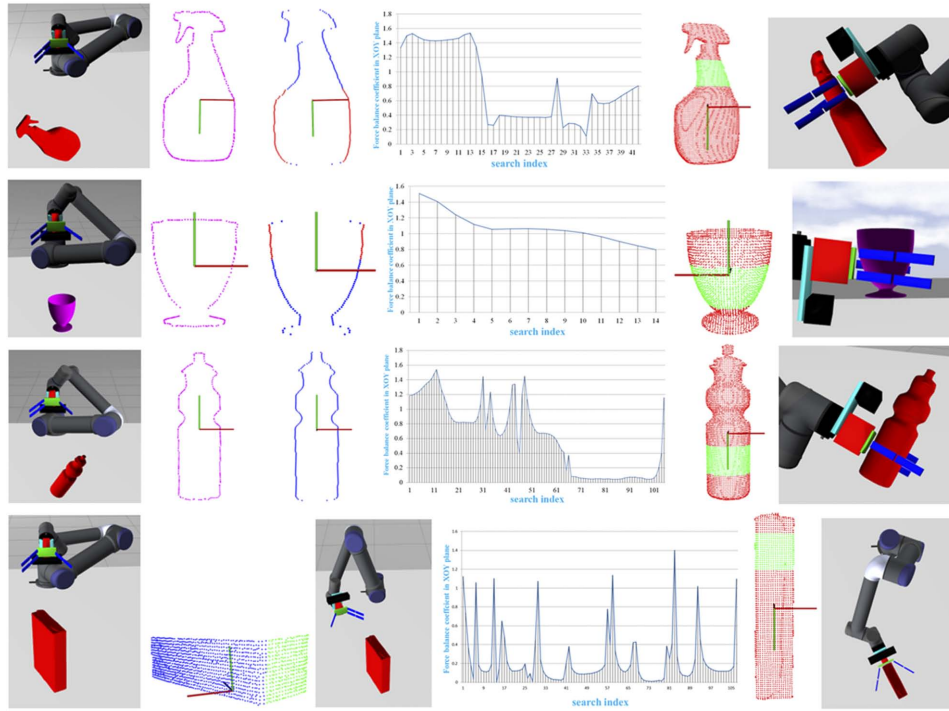


FIG. 13. Simulation results of using a single-view partial point cloud as input.

software for robot motion planning) incorporates the latest achievements in navigation, manipulation and kinetics. The first and foremost part is simulation for a single-view partial point cloud obtained using a 3D camera, which is illustrated below.

### B. Simulations based on a single-view partial point cloud using a 3D camera

The simulations of using a single-view partial point cloud as input are conducted to obtain suitable grasp. In the simulations, a spray bottle, a cup, a water bottle and an oatmeal box which have varying geometries are employed. The first column shows the simulation setup of the robot and the target object. The first three rows of Figure 13 illustrate the simulations to determine suitable grasp on a single-view partial point cloud. The last row of Figure 13 shows the method of using the main plane of the single-view partial point cloud to guide the robot to the second viewpoint. For the first three rows, the second column shows concave hull boundary of the single-view partial point acquired using one 3D camera. The blue points in the third column stand for the region with grasping range of the robot hand and this region is graspable zone. The grasp candidates are allocated from the top to the bottom of the graspable zone. Force balance computation on the XOY plane is conducted on these grasp candidates and the forth column shows the results of force balance computation. The corresponding best grasping areas are shown as green points in the fifth column. The last column demonstrates successful grasp execution for the target unknown objects. The last row demonstrates how to use the main plane of the single-view partial point cloud of the target object to guide the robot to the second viewpoint to achieve a suitable grasp.

Table I presents the detailed results of force balance computation on both the XOY plane and The XOZ plane. Based on force balance on the XOY plane, our grasping algorithm can be used to

TABLE I. Simulation results of force balance computation using a single-view partial point cloud as input.

	Spray bottle	Cup	Water bottle	Oatmeal box
XOY (radian)	0.108467	0.797198	0.0446418	0.000365332
XOZ (mm)	2.185	0.4709	0.01	1.56499

achieve stable grasping results for Spray bottle, Water bottle and Oatmeal box. For the selected cup, the force balance coefficient of the cup on the XOY plane is much larger due to its own geometrical character of no parallel sides. For force balance on the XOZ plane, results for all objects are quite satisfying. The maximum difference is 2.185 millimeter which cannot lead to massive movement of the target object when the robot tries to grasp it. As for the computing time, time spent to process a single-view partial point cloud is within one second.

In summary, using the principle axis can reasonably determine the essential feature of an unknown target. It demonstrated that this grasping algorithm can achieve a greater grasping speed to obtain a suitable grasping areas compared to others, thus the proposed algorithm is numerically verified. In order to validate this novel algorithm to practice, the experimental tests are conducted as follows.

#### IV. EXPERIMENTAL VALIDATION

In order to demonstrate the applicability of our proposed algorithm in reality, experimental tests are carried out using four different unknown objects of different shapes. The experimental tests are designed based on the references of the simulations and the availability of unknown objects.

##### A. Experimental description

An eye-in-hand system consisting of a Universal Robot arm UR5, an under-actuated Lacquey Fetch gripper and an Xtion pro live sensor is used to conduct experiment tests. The Xtion pro live

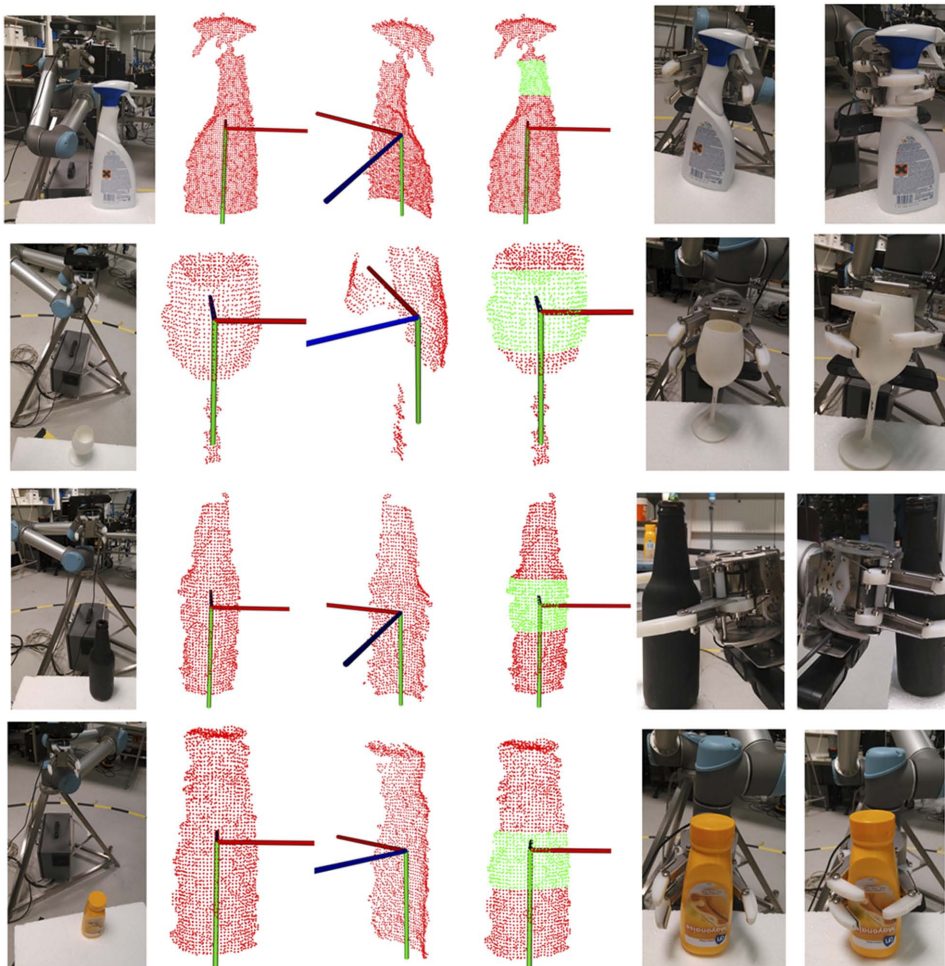


FIG. 14. Snapshots from the experimental tests.



TABLE II. Experiment results of force balance computation on the XOY plane and the XOZ plane.

	Spray bottle	Wine glass	Beer bottle	Mayonnaise bottle
XOY(radian)	0.117	0.218	0.017	0.326
XOZ(mm)	2.635	0.504	0.363	0.002

sensor and the under-actuated gripper are installed at the end of the robot arm. With the reference of simulation models, four experimental tests are carried out as shown below as shown in Figure 14.

In the Figure 14, the first column presents the whole experiment setup. The second and third columns of Figure 14 show the single-view partial point cloud. The second column corresponds to the single-view partial point cloud obtained from the perspective of the front direction and the third column corresponds to the single-view partial point cloud from the back direction. The final grasps returned from our grasping algorithm of using a single-view partial point cloud as input are shown as the fourth column of Figure 14.

## B. Results

Table II shows the experiment results of force balance computation on the XOY plane and the XOZ plane. We can find all final grasps are with a good force balance which can ensure the grasp stability. The fifth column of Figure 14 shows the under-actuated Lacquey Fetch gripper arrives at the grasping point. The last column demonstrates the successful grasp execution of the target unknown objects. Our grasping algorithm can quickly process the single-view partial point cloud of the target unknown object to output the final grasp within one second.

Thus far, our algorithm has been validated using experimental tests. It demonstrated that our fast grasping algorithm can promote successful grasping results using a single-view partial point cloud as input. In order to facilitate this novel algorithm to practice, solutions for minimizing grasping uncertainty is illustrated in the next section.

## V. MINIMIZING GRASPING UNCERTAINTY BY USING TWO 3D CAMERAS

As was illustrated in section II, using geometric properties of the target unknown object to obtain suitable grasps may bring about the risk of grasping failure as a result of the uncertainties induced from occlusions. For these scenarios, the robot has to move to obtain new perspective to obtain single-view point cloud data. In practice, the robots that have two cameras can be used to overcome grasp uncertainty resulting from occlusions. This can be achieved by following the six (A-F) procedures.

### A. Building a “big” partial point cloud using two 3D cameras

Figure 15 shows a Baxter robot and PR2 robot which have two camera sensors separately installed at the robot head and robot hand. Using these types of robots, more completed contour

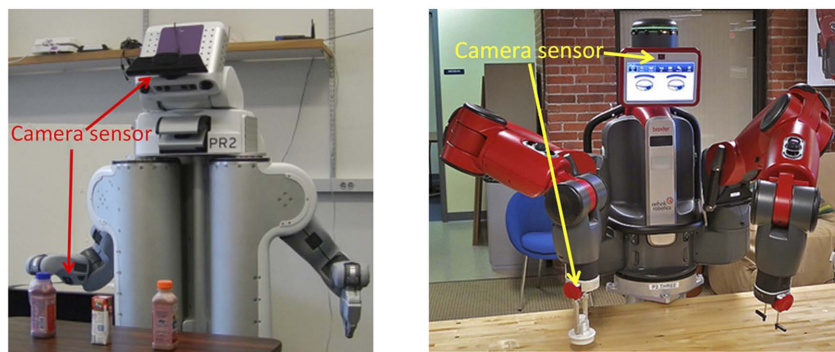


FIG. 15. Two example robots which use two camera sensors.

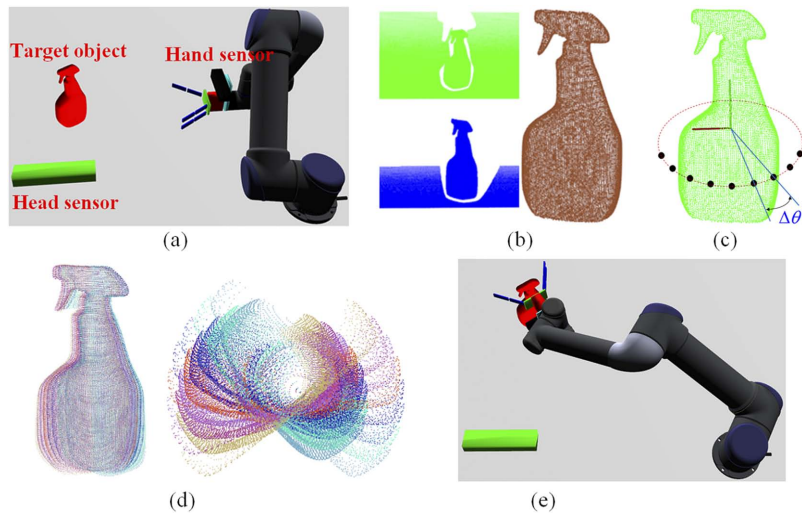


FIG. 16. The method to overcome grasp uncertainty resulting from occlusions. (a) PR2 and Baxter robots are simplified as a head sensor and a hand sensor (b) two single-view partial point cloud from the two 3D sensors fused together (c) Virtual viewpoints incrementally allocated around the principal axis of the “big” partial point cloud. (d) the corresponding point clouds for all virtual viewpoints (e) an example of successful grasp execution.

of the target unknown object can be formed compared to single-view partial point cloud. After that, virtual exploration is carried out on the “big” partial point cloud to carry out searching of principle axis. Graspable candidates are allocated between the two camera sensor points, which can ensure the grasp candidates are allocated on the seen part of the target object. While constructing the “big” partial point cloud cannot result in much computing time, the increase of grasp security is achieved.

The PR2 and Baxter robots in Figure 15 can be simplified as a head sensor and a hand sensor. The two 3D sensors are used to obtain the “big” partial point cloud. The green and blue point cloud in Figure 16 (b) respectively stand for a single-view partial point cloud acquired from the head sensor and the hand sensor. Next, the two single-view partial point clouds are fused together to obtain the “big” partial point cloud (shown as the brown point cloud in Figure 16 (b)). When we obtained the “big” partial point cloud, virtual viewpoints (shown as the black points in Figure 16 (c)) can be incrementally allocated around the principal axis between the head sensor point and the hand sensor point. Figure 16 (d) shows the corresponding point clouds for all virtual viewpoints.

Figure 17 illustrates the distribution of the camera sensors of the Baxter robot and the PR2 robot to obtain “big” partial point cloud. The green cuboid stands for the camera on the robot head and the black cuboid at the end of the robot arm represent the camera on the robot arm. The coordinate transformation between the head sensor coordinate system (HESCS) and the base link coordinate system (BCS) is defined as ( $T_{head}$ ). The coordinate transformation between the hand sensor

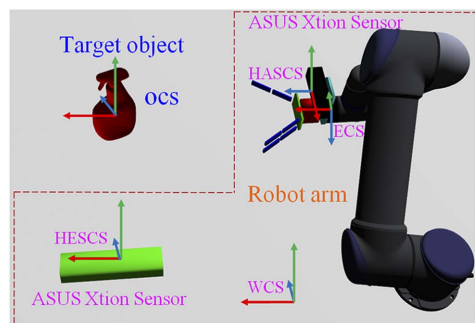


FIG. 17. Coordinate transformation between two camera sensors and the world coordinate system (WCS).

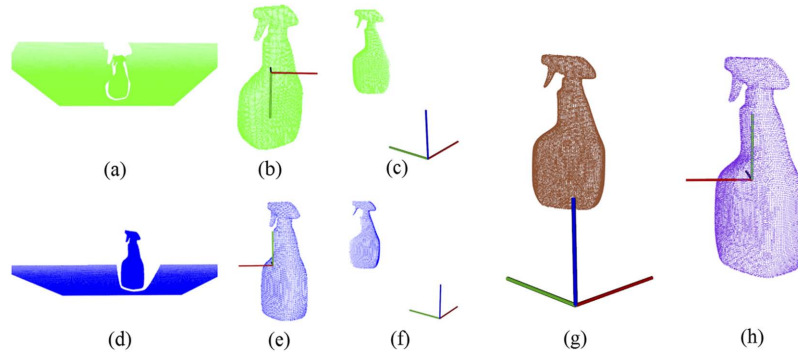


FIG. 18. Acquisition of the “big” partial point cloud and transfer it from the world coordinate system (WCS) to the Object coordinate system (OCS). (a) and (d) are the single-view partial point cloud obtained from the head camera sensor and hand camera sensor. (b) and (e) are the two single-view partial point clouds of the target object in the camera coordinate system after removing the table plane. (c) and (f) are the two single-view partial point clouds transferred into world coordinate system (WCS). (g) is the registered “big” partial point cloud in the WCS. (h) is a “big” partial point cloud transferred from WCS to object coordinate system (OBS).

coordinate system (HASCS) and the base link coordinate system (BCS) is defined as  $(T_{hand}) \cdot T_{head} \cdot T_{hand}$  can be obtained by equation (7).

$$\begin{cases} T_{head} = T_{HESCS\_BCS} \\ T_{hand} = T_{HASCS\_ECS} * T_{ECS\_BCS} \end{cases} \quad (7)$$

Figure 18 shows the procedures to acquire the “big” partial point cloud and transfer it from the world coordinate system (WCS) to the Object coordinate system (OCS). Figure 18(a) and (d) respectively stand for the single-view partial point cloud obtained from the head camera sensor and hand camera sensor. Figure 18(b) and (e) stands for the two single-view partial point clouds of the target object in the camera coordinate system after removing the table plane. Figure 18(c) and (f) stand for the two single-view partial point cloud transferred into the world coordinate system (WCS). Figure 18(g) is the registered “big” partial point cloud in the WCS. Figure 18(h) shows the “big” partial point cloud transferred from WCS to object coordinate system (OBS).

## B. Grasp allocation between the two camera sensor points

Figure 19 illustrates the searching strategies between the two camera sensor points. In this figure, each black point stands for a virtual viewpoint to carry out virtual exploration. The virtual viewpoints are allocated around the principal axis with a searching step  $(\Delta\theta)$ .  $q_{\theta 0}$  and  $q_{\theta 1}$  stand for the hand and the head camera viewpoints, respectively.  $q_{\theta i}$  means the  $i_{th}$  virtual viewpoint and  $q_{\theta i} = q_{\theta 0} + i * \Delta\theta$ .

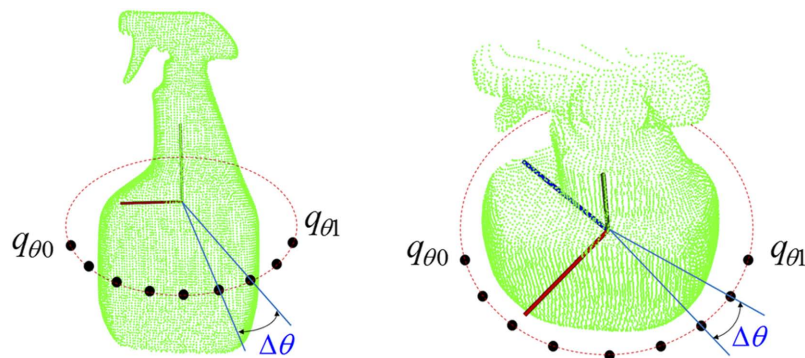


FIG. 19. Searching strategies between the two camera sensor points.



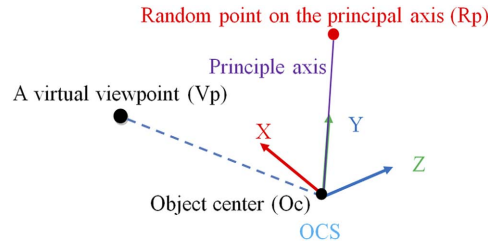


FIG. 20. Schematic illustration of constructing a virtual object coordinate system (VOCS).

### C. Constructing virtual object coordinate systems

At each virtual viewpoint, a virtual object coordinate system (VOCS) is constructed for further analysis. Figure 20 illustrates the construction of the virtual object coordinate system. Specifically, the principal axis of the “big” partial point cloud in WCS is used to work as the Y axis of the object coordinate system (OCS). In Figure 20, Rp stands for a random point on the principal axis and Vp means a random virtual viewpoint. The X and Z axis can be obtained using equation (8).

Using the above method to go through every virtual viewpoint of Figure 19, we can obtain all VOCSs for all virtual viewpoints. Then the “big” partial point cloud in WCS is transferred to every local VOCS, all the transferred point clouds can be seen in Figure 21. Each color in Figure 21 corresponds to a transferred point cloud at the local virtual object coordinate system.

$$\begin{cases} \overrightarrow{O_c Y} = \overrightarrow{O_c R_p} \\ \overrightarrow{O_c X} = \overrightarrow{O_c V_p} \times \overrightarrow{O_c R_p} \\ \overrightarrow{O_c Z} = \overrightarrow{O_c X} \times \overrightarrow{O_c Y} \end{cases} \quad (8)$$

### D. Grasp allocation for a virtual viewpoint

The grasp configuration in the SE(3) group means many possibilities. To reduce grasp possibility to accelerate grasp searching, principal axis of the “big” partial point cloud is used to direct the grasp configuration. In Figure 22, the grasp configuration in the SE(2) group is simply a rotation around the principal axis. Each configuration can be expressed as  $\Gamma[q_{y_i}|q_{\theta_i}]$ , in which  $y_i$  denotes grasp searching from the top of the “big” partial point cloud to the bottom of the “big” partial point cloud;  $\theta_i$  means grasp searching between the two camera sensor points.

All transferred point clouds for all VOCSs were already obtained shown in Figure 21. For each transferred point cloud  $O[q_{\theta}]$ , grasp candidates are allocated from the top to the bottom, as can be seen in Figure 23. Let  $O[q_{y_i}|q_{\theta}]$  ( $i = 1, 2, \dots, n$ ) stand for the point cloud covered by  $i_{th}$  hand configuration, such that the red, blue and green rectangles respectively stand for three hand configurations. The

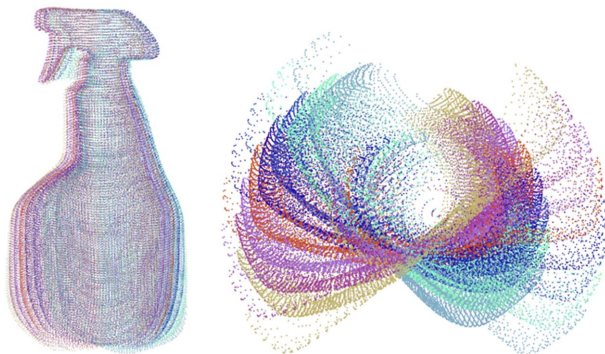


FIG. 21. Transferred point clouds for all the virtual coordinate system (VOCSs).

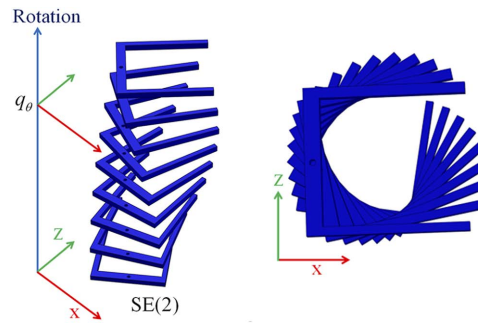


FIG. 22. Grasping configuration in the SE(2) group.

corresponding point cloud covered by the three hand configurations can be expressed as  $O[q_{yi}|q_\theta]$ ,  $O[q_{yi+1}|q_\theta]$  and  $O[q_{yi+2}|q_\theta]$ .

### E. Force balance computation

For the point cloud  $O[q_\theta]$  which corresponds to a virtual viewpoint, force balance computation can be divided into two parts, namely, force balance computation on the XOY plane ( $F_{xoy}$ ) and on the XOZ plane ( $F_{xoz}$ ). The application of using force balance to obtain the best grasping areas consisting of the following two steps.

The first step is using force balance computation on the XOY plane to select the best grasp for each virtual point cloud  $O[q_{\theta h}]$  from top to bottom, i.e.,  $F_{XOY} = \text{MAX}\{F_{XOY}\{O[q_{yi}|q_\theta] \wedge O[q_{yi}|q_\theta] \in O[q_\theta]\}\}$ . For every virtual viewpoint  $\theta$ , force balance computation on the XOY plane is used to find the best grasp, and  $y_i (i=1,2,3,\dots,n)$  stands for the  $i$ th grasp candidate allocated along the Y axis.

Next step is using force balance computation on the XOZ plane to compare the best grasping areas deduced from the first step from left to right, i.e.,  $F_{XOZ} = \text{MAX}\{F_{XOZ}\{O[q_{ym}|q_{\theta h}]\}\}$ , in

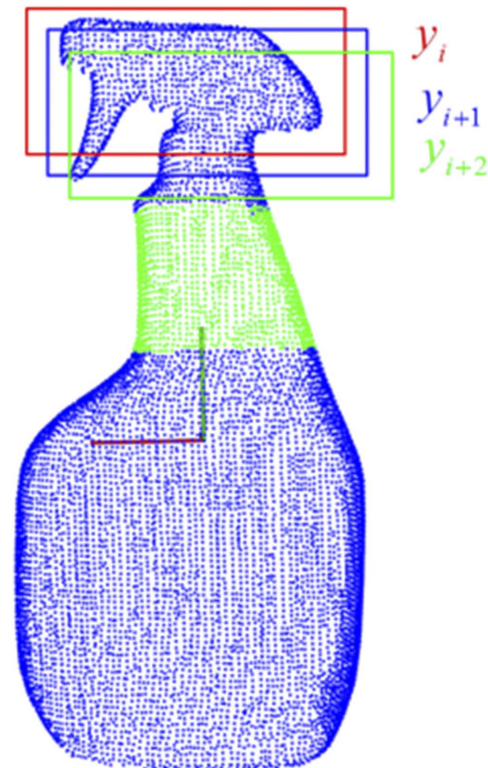


FIG. 23. Hand configuration for a virtual viewpoint.

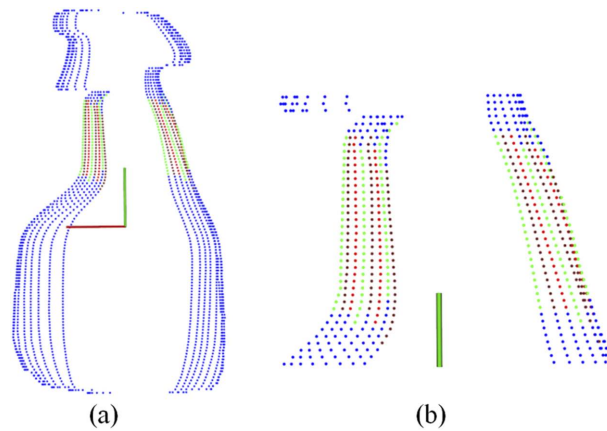


FIG. 24. Results of force balance computation on the XOY plane for all virtual viewpoints. (a) best grasping areas for all virtual viewpoints (b) partial enlarged image.

which  $\theta h$  ( $h = 1, 2, 3, \dots, n$ ) stands for the  $h$ th virtual viewpoint.  $O[q_{\theta h}]$  stands for the best grasp candidate of the  $h$ th virtual viewpoint. Force balance computation on the XOZ plane is carried out on the best grasping areas of every virtual viewpoint to choose the best grasp as final grasp execution.

### 1. Force balance computation on the XOY plane for each virtual viewpoint

As explained previously, force balance computation is firstly carried out to determine the best grasping areas for each virtual viewpoint. For a virtual viewpoint, the parameter ( $\theta$ ) corresponds to a virtual point cloud  $O[q_{\theta}]$ . The method in the step 6 of section II.A is used to find the best grasp for the virtual point cloud  $O[q_{\theta}]$  on the XOY plane. Using the above method for going through all the virtual viewpoints, we can obtain all the best grasping areas for all virtual viewpoints. These best grasping areas can construct a grasp vector  $G = (g_1, g_2, \dots, g_n)$ , and  $g_n$  stand for the best grasp for the  $n$ th virtual viewpoint, and  $n$  means the total number of virtual viewpoints. Best grasping areas for all virtual viewpoints are shown as Figure 24. In the next subsection, the selection of the final grasp from the grasp vector  $G = (g_1, g_2, \dots, g_n)$  is illustrated.

### 2. Force balance computation on the XOZ plane for every virtual viewpoint

In order to choose the final grasp from the grasp vector  $G = (g_1, g_2, \dots, g_n)$ , the best grasping areas for every virtual viewpoint are extracted shown as Figure 25 (a) and (b). The colorful points in the black circle stand for all the grasps in the vector  $G = (g_1, g_2, \dots, g_n)$ . Figure 25 (c) shows an example grasp  $g_i$  ( $1 \leq i \leq n$ ) in the vector  $G_j$ . The green points in Figure 25 (c) stand for points covered the  $g_i$ . To evaluate the grasp quality of  $g_i$ , force balance computation on the XOZ plane can be used. In order to compute force balance on the XOZ plane, the grasp  $g_i$  is first projected to the XOY plane to

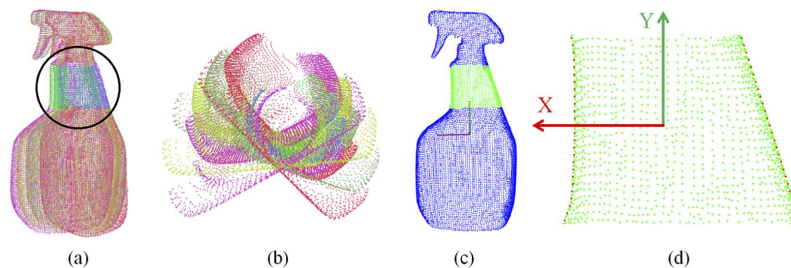


FIG. 25. Best grasping areas for every virtual viewpoint are extracted to do force balance computation on the XOZ plane. (a) and (b) are the best grasping areas of all virtual viewpoints are extracted. Green points in (c) and (d) stand for an example of the achieved grasping area.

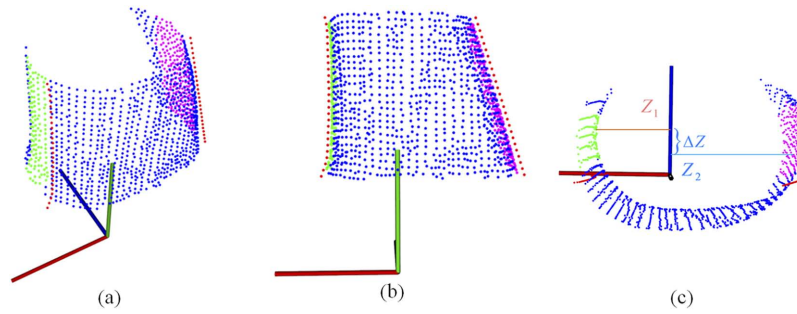


FIG. 26. The difference between the left and the right contact region of the grasp is used to evaluate the grasp stability. (a) and (b) extracted left and right contact regions. (c) the average Z values of the left and right contact region.

obtain the most left and most right contact area shown as the red points in Figure 25 (d). The left red points is numbered as  $p_{l1}, p_{l2}, \dots, p_{lm}$  ( $m$  is the total number of left red points). The right red points is numbered as  $p_{r1}, p_{r2}, \dots, p_{rn}$  ( $n$  is the total number of right red points).

$$\begin{cases} (X_{lk} \leq X_i \leq X_{l(k+1)}) \parallel (X_{l(k+1)} \leq X_i \leq X_{lk}) \\ (X_{rk} \leq X_i \leq X_{r(k+1)}) \parallel (X_{r(k+1)} \leq X_i \leq X_{rk}) \end{cases} \quad (9)$$

Using equation 9, we can obtain the left and right contact area between the robot hand and the target object. By so doing a point of the grasp  $g_i$  is extracted when the X value of this point is located between two adjacent red points as shown in Figure 25 (d). The extracted left and right contact regions between the robot hand and the target object are shown as green points and purple points in Figure 26 (a) and (b). Then average Z values of the left and right contact region are worked out shown as  $Z_1$  and  $Z_2$  shown as Figure 26 (c). The difference ( $\Delta Z$ ) between  $Z_1$  and  $Z_2$  is used to evaluate the stability of this grasp (shown as equation (10)). It is obvious to find that large difference ( $\Delta Z$ ) will lead to rotation of the object around the Y axis, which may lead to grasp failure. Using above method goes through all the grasp of vector  $G = (g_1, g_2, \dots, g_n)$ , we can obtain the force balance on the XOZ plane for all grasp candidates of vector  $G$ , that is vector  $\Delta Z = (\Delta Z_1, \Delta Z_2, \dots, \Delta Z_n)$ . A line graph (Figure 27) is drawn according to the vector  $\Delta Z = (\Delta Z_1, \Delta Z_2, \dots, \Delta Z_n)$ . We can predict that force balance on the XOZ plane reaches the best when virtual viewpoint  $\theta_i$  comes to  $\theta_4$ . Therefore, the fourth grasp  $g_4$  of the vector  $G = (g_1, g_2, \dots, g_n)$  is chosen as final grasp execution.

$$\Delta Z = |\bar{Z}_{left} - \bar{Z}_{right}| = \left| \frac{1}{m} \sum_{i=1}^m z_i - \frac{1}{n} \sum_{i=1}^n z_i \right| \quad (10)$$

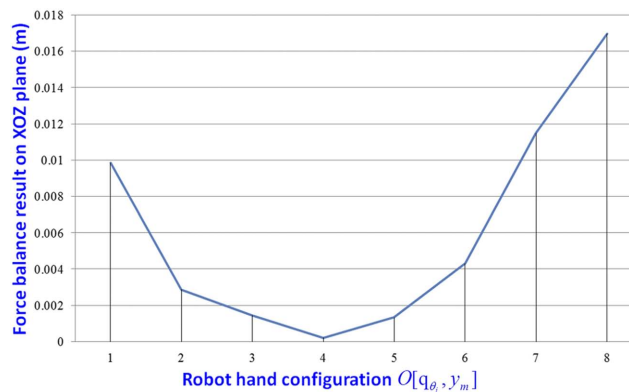


FIG. 27. Results of force balance computation on the XOZ plane for all best grasping areas of every virtual viewpoint.

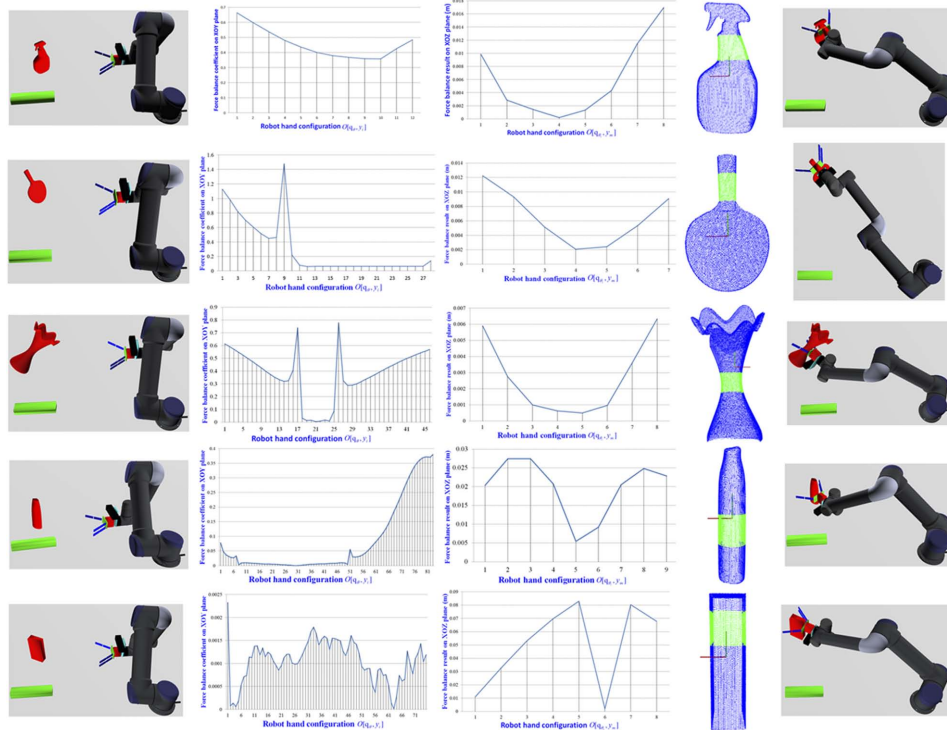


FIG. 28. Simulation results of “big” partial point cloud obtained using two 3D cameras.

### F. Simulation for the “big” partial point cloud obtained by using two 3D sensors

In this subsection, simulations of using two 3D cameras to construct a “big” partial point cloud are conducted to obtain suitable grasping areas. Inspired by the Baxter and PR2 robots, we set two 3D cameras in the simulation setup. The black and green cuboids in the first column of Figure 28 respectively stand for the hand camera and head camera. Objects used to conduct simulations can be seen in the first column of Figure 28. Two single-view partial point cloud of the target object from the two 3D cameras are used to construct a “big” partial point cloud. Force balance computation is conducted on the “big” partial point cloud and the results of force balance computation are shown as the second column and the third column of Figure 28.

Detailed results of force balance computation can be seen in Table III. It demonstrates that appropriate force balances for all tested objects are obtained on both the XOY plane and the XOZ plane. The fourth column of Figure 28 shows the final grasp returned from our grasping algorithm.

TABLE III. Final grasp configuration and the corresponding force balance coefficient for the “big” partial point cloud obtained by using two 3D cameras.

Force balance 1		Force balance 2		Force balance 3		Force balance 4		Force balance 5	
XOY	XOZ	XOY	XOZ	XOY	XOZ	XOY	XOZ	XOY	XOZ
0.3576	0.0002	0.0618	0.0021	0.0034	0.0005	0.0003	0.0054	8.5e-6	0.0021



TABLE IV. Grasp computing time of using two 3D cameras to construct a “big” partial point cloud.

Unknown Objects	Spray bottle	Table tennis bracket	Vase	Shampoo bottle	Oatmeal box
points	9801	1358	19583	7059	11214
Time (s)	1.068	0.515	1.653	0.994	1.198

The last column demonstrates successful grasp execution for all target unknown objects. The fourth row of Table III shows the final grasps for all tested unknown objects. The third row of Table III shows the simulation results of force balance computation on both the XOY plane and the XOZ plane. We can find all the final grasps of the target unknown objects show expected force balance performances on both the XOY plane and XOZ plane. Thus, it ensures the stability of final grasp execution for all the tested unknown objects.

Table IV shows the grasp computing time of our grasping algorithm using a “big” partial point cloud. Even though the point cloud is based on a large number of points, our grasping algorithm can quickly process the “big” partial point cloud and output the final grasp within two seconds. This result demonstrates a faster grasping speed compared to the other researchers that use multi cameras<sup>30</sup> or multi views.<sup>31–33</sup>

To sum up, the simulation tests to predict the grasping effectiveness in realistic situations are carried out using two 3D cameras. It demonstrated that the risk of grasping failures can be highly reduced in comparison with the application of single-view partial point cloud. While the grasping efficiency is maintained, our algorithm shows the improvements for robots to fast grasp unknown objects under unpredictable environments.

## VI. CONCLUSION

A novel algorithm of using principal component analysis for fast grasping unknown objects is proposed. For a single-view partial point cloud, graspable candidates are allocated along the principal axis from the top of the target object to the bottom of the target object. Force balance computation on both the XOY plane and the XOZ plane ensures the stability of the final grasping action. To illustrate the efficiency of our grasping algorithm, objects with different geometric shapes are used to conduct simulations and experiments. Both simulation and experimental tests demonstrated favorable performances of applying the algorithm. In addition, it shows that using our grasping algorithm the speed of grasping is greater than other algorithms for steady grasping.

In order to facilitate this algorithm in practice, the grasping uncertainty is minimized by taking the advantage of two cameras of the robot hardware. Virtual exploration on the “big” partial point cloud is carried out to determine the final grasp with the best force balance. The simulation results demonstrated that our grasping algorithm can quickly accomplish virtual exploration with steady grasping result. Therefore, this research demonstrates practical significance for increasing grasping speed and thus increasing robot efficiency under unpredictable environments.

## ACKNOWLEDGMENTS

This research was supported by the project: Factory in a day, from European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 609206.

<sup>1</sup> J. Meijer, Q. Lei, and M. Wisse, “An empirical study of single-query motion planning for grasp execution,” 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM 2017), 59–516, (2017)

<sup>2</sup> J. Bohg, A. Morales, T. Asfour, and D. Kragic, “Data-driven grasp synthesis: A survey,” IEEE Transactions on Robotics **30**(2), 1–21 (2013).

<sup>3</sup> J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergstrom, D. Kragic, and A. Morales, “Mind the gap-robotic grasping under incomplete observation,” 2011 IEEE International Conference on Robotics and Automation (ICRA), 686–693, (2011).

<sup>4</sup> 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* **107**, 146–151 (2003).

<sup>5</sup> Y. Jiang, S. Moseeson, and A. Saxena, “Efficient grasping from RGBD images: Learning using a new rectangle representation,” IEEE International Conference on Robotics and Automation, 3304–3311, (2011).

- <sup>6</sup> Y.-C. Lin, S.-T. Wei, and L.-C. Fu, "Grasping unknown objects using depth gradient feature with eye-in-hand RGB-D sensor," 2014 IEEE International Conference on Automation Science and Engineering (CASE), 1258–1263, (2014).
- <sup>7</sup> M. Popovic, G. Kootstra, J. A. Jorgensen, D. Kragic, N. Kruger, J. A. Jørgensen, and N. Krueger, "Grasping unknown objects using an early cognitive vision system for general scene understanding," In: International conference on intelligent robots and systems (IROS), 987–994, (2011).
- <sup>8</sup> J. Baumgartl and D. Henrich, "Fast vision-based grasp and delivery planning for unknown objects," 7th German Conference on Robotics (ROBOTIK 2012), 1–5, (2012).
- <sup>9</sup> M. Richtsfeld and M. Vincze, "Grasping of unknown objects from a table top," Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments, (2008).
- <sup>10</sup> L. Bodenhausen, D. Kraft, M. Popovic, E. Bašeski, P. Eggenberger Hotz, and N. Krüger, "Learning to grasp unknown objects based on 3D edge information," Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA, 421–428, (2009).
- <sup>11</sup> R. K. Ala, D. H. Kim, S. Y. Shin, C. H. Kim, and S.-K. Park, "A 3D grasp synthesis algorithm to grasp unknown objects based on graspable boundary and convex segments," *Information Sciences* **295**, 91–106 (2015).
- <sup>12</sup> A. Maldonado, U. Klank, and M. Beetz, "Robotic grasping of unmodeled objects using time of flight range data and finger torque information," IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010-Conference Proceedings, 2586–2591, (2010).
- <sup>13</sup> A. T. Pas and R. Platt, "Using geometry to detect grasps in 3D point clouds," 2015 International Symposium on Robotic Research. 1–16 (2015).
- <sup>14</sup> S. E. Navarro, "Tracking and grasping of known and unknown objects from a conveyor belt," The 41st International Symposium on Robotics. (2014).
- <sup>15</sup> T. Suzuki and T. Oka, "Grasping of unknown objects on a planar surface using a single depth image," 2016 IEEE International Conference on Advanced Intelligent Mechatronic (AIM 2016), 572–577 (2016)
- <sup>16</sup> B. Calli, M. Wisse, and P. Jonker, "Grasping of unknown objects via curvature maximization using active vision," In: International conference on intelligent robots and systems (IROS), 995–1001, (2011).
- <sup>17</sup> J. Bao, Y. Jia, Y. Cheng, and N. Xi, "Saliency-guided detection of unknown objects in RGBD indoor scenes," *Sensors* **15**(9), 21054–21074 (2015).
- <sup>18</sup> J. Felip and A. Morales, "Robust sensor-based grasp primitive for a three-finger robot hand," In: International conference on intelligent robots and systems (IROS), 1811–1816, (2009).
- <sup>19</sup> R. Platt, A. H. Fagg, and R. A. Grupen, "Nullspace composition of control laws for grasping," In *International conference on intelligent robots and systems (IROS)*, 1717–1732, (2002).
- <sup>20</sup> M. Prats, P. Martinet, S. Lee, and P. J. Sanz, "Compliant physical interaction based on external vision-force control and tactile-force combination," In IEEE International Conference on Multisensor Fusion & Integration for Intelligent Systems (MFI), 405–410, (2008).
- <sup>21</sup> K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones, "Contact-reactive grasping of objects with partial shape information," In: International conference on intelligent robots and systems (IROS), 1228–1235, (2010).
- <sup>22</sup> K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," In IEEE International Conference on Robotics and Automation (ICRA), 2098–2105, (2009).
- <sup>23</sup> S. Dragiev, M. Toussaint, and M. Gienger, "Uncertainty aware grasping and tactile exploration," In IEEE International Conference on Robotics and Automation (ICRA), 113–119, (2013).
- <sup>24</sup> C. Zito, M. S. Kopicki, R. Stolkin, C. Borst, F. Schmidt, M. Roa, and J. L. Wyatt, "Sequential trajectory re-planning with tactile information gain for dexterous grasping under object-pose uncertainty," In: International conference on intelligent robots and systems (IROS), 4013–4020. (2013).
- <sup>25</sup> N. Sommer, M. Li, and A. Billard, "Bimanual compliant tactile exploration for grasping unknown objects," IEEE International Conference on Robotics and Automation, 6400–6407, (2014).
- <sup>26</sup> Z. Yi, R. Calandra, F. Vegia, H. van Hoof, T. Hermans, Y. Zhang, and J. Peters, "Active tactile object exploration with Gaussian processes," In: International conference on intelligent robots and systems (IROS), 4925–4930, (2016).
- <sup>27</sup> A. Molchanov, O. Kroemer, Z. Su, and G. S. Sukhatme, "Contact localization on grasped objects using tactile sensing," In: International conference on intelligent robots and systems (IROS), 216–222, (2016).
- <sup>28</sup> C. Eppner and O. Brock, "Grasping unknown objects by exploiting shape adaptability and environmental constraints," In proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4000–4006, (2013).
- <sup>29</sup> Z. Liu, H. Kamogawa, and J. Ota, "Fast grasping of unknown objects through automatic determination of the required number of mobile robots," *Advanced Robotics* **27**(6), 445–458 (2013).
- <sup>30</sup> G. M. Bone, A. Lambert, and M. Edwards, "Automated modelling and robotic grasping of unknown three-dimensional objects," 2009 IEEE International Conference on Robotics and Automation (ICRA), 292–298, (2008).
- <sup>31</sup> A. Collet and S. S. Srinivasa, "Efficient multi-view object recognition and full pose estimation," 2010 IEEE International Conference on Robotics and Automation (ICRA), 2050–2055, (2010).
- <sup>32</sup> G. Kahn, P. Sujan, S. Patil, S. Bopardikar, J. Ryde, K. Goldberg, and P. Abbeel, "Active exploration using trajectory optimization for robotic grasping in the presence of occlusions," 2015 IEEE International Conference on Robotics and Automation (ICRA), 4783–4790, (2015).
- <sup>33</sup> Y. Hu, G. Lin, C. Yang, Z. Li, and C.-Y. Su, "Manipulation and grasping control for a hand-eye robot system using sensory-motor fusion," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), 351–356, (2015).