applied sciences

MDPI

Article

Self-Recognition Grasping Operation with a Vision-Based Redundant Manipulator System

Tong Li [1,*], Shuaikang Zheng [1,2], Xin Shu [1,2], Chunkai Wang [1,2] and Chang Liu [1,2]

1 State Key Laboratory of Transducer Technology, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China; zhengshuaikang18@mails.ucas.ac.cn (S.Z.); shuxin16@mails.ucas.ac.cn (X.S.); wangchunkai16@mails.ucas.ac.cn (C.W.); changliu8888@gmail.com (C.L.)

2 School of Electronic, Electrical, and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China

* Correspondence: tli@mail.ie.ac.cn

Received: 5 November 2019; Accepted: 26 November 2019; Published: 28 November 2019

Abstract: For unstructured environment applications, the ability of self-recognition for grasping operations should be guaranteed for manipulators. For this purpose, a grasping process, including instance segmentation, pose estimation, and pose transformation, is proposed herein to achieve autonomous object detection, location detection, and grasp planning. An inverse solution in position form is derived for pose transformation to guarantee redundant manipulator adaption. The inverse solution requires no default initial configuration and can obtain all feasible solutions for grasping. Additionally, the optimal grasp can be selected by introducing an optimal factor, such as manipulability. Besides, the process is programmed with high computational efficiency, making it a better choice for manipulators to achieve self-recognized grasping operation. Experiments are carried out herein to verify the necessity of instance segmentation, pose estimation, and pose transformation in achieving self-recognized grasping operation. The inverse solution in the position form is also proven to be efficient and adaptable for the pose transformation of redundant manipulators.

Keywords: redundant manipulator; self-recognition grasping operation; instance segmentation; pose transformation; vision

1. Introduction

Nowadays, manipulators have an increasing degree of requirement in many fields, since grasping operations [1,2] with manipulators can be competent for various works that substitute human beings. A redundant manipulator which possesses more DOF (degree of freedom) and dexterity in three-dimensional space has a better performance in self-perception and self-learning when in an unstructured environment and amongst clutter. The ability of autonomy is increasingly becoming more and more important for manipulators in complex scene applications. Thus, it is significant for redundant manipulators to possess the ability of self-recognition, especially for grasping operations.

To achieve self-recognized grasping operation, redundant manipulators need to complete object detection, object location, and grasp planning autonomously. In object detection, the target object is distinguished from others and the environment. In terms of the object location, the 3D (three dimensional) position and orientation angle of the target object are obtained in a pixel coordinate system. Then, via pose transformation, a joint angle sequence is derived for manipulator control via the pose (position and orientation) expressed in the pixel coordinate system.

Vision systems which can obtain abundant information about the present environment have already been integrated in manipulator grasping operation [3,4]. In the industry field, templates of objects (components or work piece) are made in advance. Features such as SIFT (scale-invariant

feature transform) [5] and SURF (speeded-up robust features) [6] are extracted from the image to achieve shape-matching [7]. Since the position of an object is traceable in the workbench, methods such as the optical flow method [8] are applied for simplified location and motion tracking. However, these methods rely on the known classification and position of objects, and they cannot be adapted for unstructured environments and those where clutter objects with unknown information exist. Indeed, with vision systems, target object detection and location can be achieved preferentially for self-recognition grasping operation.

Benefit from the studies on deep learning, CNNs [9,10] (convolutional neural networks) have been used to achieve object recognition, leading to various methods such as image classification [11,12], object detection [13,14], semantic segmentation [15,16], and instance segmentation [17,18]. Many frameworks of instance segmentation have been proposed, such as SDS (simultaneous detection and segmentation) [19], CFM (convolutional feature masking) [20], MNC (multi-task network cascades) [21], and so on. Mask R-CNN (regions with CNN features) [22], which achieves object classification and contour description simultaneously, can accurately determine target objects from an unstructured environment. Besides, the contour has a smaller region than the classification box for object pose estimation.

After obtaining the contour and classification of an object by instance segmentation, the location of the object should be determined. In traditional methods, an RGB image is used to achieve pose estimation for the target object by matching local features [23,24]. However, this is worse in performance when locating textureless objects. However, RGB-D images, which describe the depth information with a point cloud, can directly obtain the 3D position objects [25–27]. The benefit from grasping databases [28–30] and orientation estimation is converged into an classification problem [31] via discretizing orientation angles. As a result, the pose of target object can be located in the pixel coordinate system.

Pose transformation can be divided into two steps. Firstly, the pose expressed in pixel coordinates is transferred into Cartesian coordinates via the parameters of the camera, which is explained in [32,33]. Second, the pose is transferred into the joint space of the manipulator via inverse kinematics. The inverse solution problem [34] is an old question for robotics, which can be solved in either the velocity or position forms. The inverse solution in velocity form (ISVF) relies on the Jacobian matrix of the manipulator, which represents the relationship between the joint velocity and the velocity of the end-effector. In ISVF, an initial configuration should be set, then simulated, where planning is carried out from the initial configuration to the pose of the object. The angles of each joint are converged by an integral of joint velocity at each motion interval. The process is fussy and costly in time, making ISVF suitable for offline trajectory planning.

The inverse solution in position form (ISPF) can obtain joint angles directly, and can generally be divided into numerical [35] and geometric methods [36]. However, the numerical method presents an abundant computation cost, leading to poor real-time adaption. Additionally, it often falls into a local optimum. The geometric method is only effective for low DOF manipulators (i.e., those not exceeding 6-DOF). For manipulators with a spatial redundancy (i.e., those with at least 7-DOF), this method is not compatible. Indeed, there are infinite solutions for a particular pose with redundant manipulators (DOF > 6). Less methods can achieve the expression of all feasible solutions, rapid calculation, and optimal solution selection simultaneously, which are imperative for pose transformation in self-recognized grasping operation. Thus, a method of ISPF, which can express the whole solution set rapidly and build up a rule selecting the global optimal solution, is significant for redundant manipulator application in unstructured environments and environments with clutter.

The contribution of the paper is exposing the process of self-recognition grasping operation with a vision-based spatially redundant manipulator. Instance segmentation is used to distinguish the object and describe the contour. Within the contour, pose estimation is carried out via a grasping network. On this basis, a method is proposed to calculate the inverse solution in position form to transfer the pose into the joint space of the manipulator for direct motion control.

The Section 2 shows the process of the vision-based self-recognized grasping operation, including the framework of instance segmentation, the grasping network of pose estimation, and the methods of pose transformation. In the Section 3, the inverse method of ISPF is described in detail. In the Section 4, experiments on self-recognition grasping operation are carried out. The process of self-recognition is verified and the method of ISPF is proven to be effective for real-time calculation and global optimal solution selection. The Section 5 presents our conclusion.

2. Self-Recognized Grasping Operation

For self-recognized grasping operation, the core problem is achieving object detection, pose estimation, and pose transformation autonomously. In this section, the process of the self-recognized grasping operation is described. Instance segmentation is used to classify the target object and describe the contour via a RGB-D image. During the region of the contour, pose estimation is used to locate the pose of target object in pixel coordinate system. Then, pose transformation is applied to transfer the pose, expressed in pixel coordinates, into joint angles of the manipulator. Considering the internal parameters of camera, the pose is preferentially transferred from pixel coordinates into Cartesian coordinates, then transferred into the joint space of the manipulator for direct motion control. The process of the self-recognized grasping operation is shown in Figure 1.

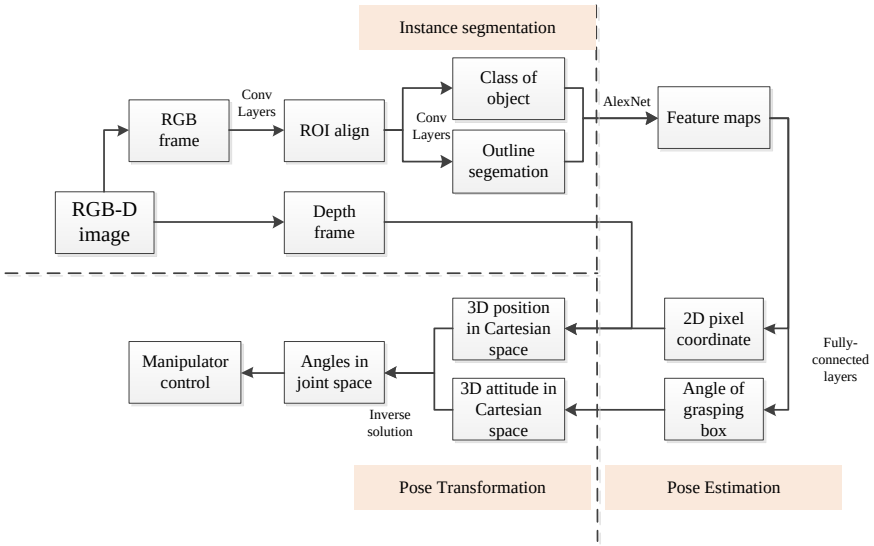


Figure 1. Process of the vision-based self-recognized grasping operation.

In the self-recognized grasping operation, instance segmentation and pose estimation rely on the RGB-D image obtained by the vision system, using CNN frameworks to achieve self-recognition. The frameworks are described in detail in Figure 2.

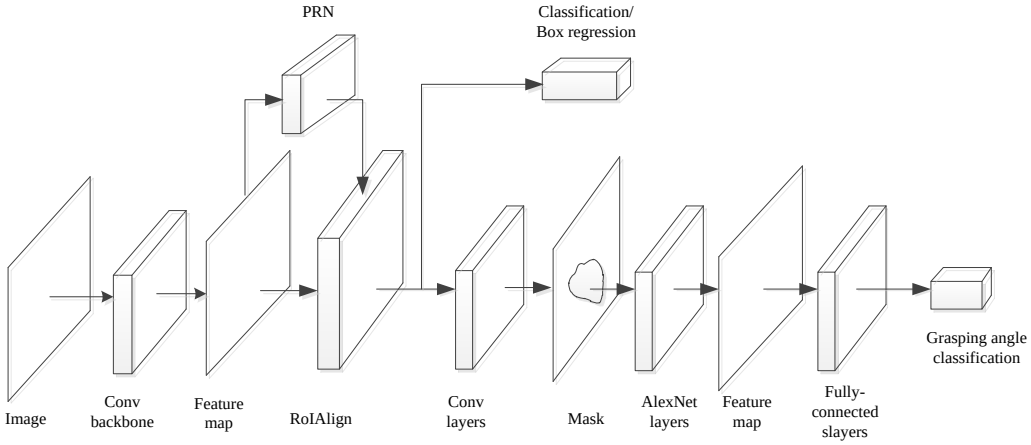


Figure 2. Framework of instance segmentation and pose estimation.

2.1. Framework of Instance Segmentation

In the framework of instance segmentation, the Fast R-CNN network is referenced. In order to improve the feature extraction and information mining, ResNet-101 [19] and feature pyramid networks [37] (FPN) are introduced as convolutional backbones to guarantee feature detection in multi-scales, where the feature map is then extracted. A region proposal network (RPN) is applied to propose candidate object bounding boxes, and a region of interest (RoI) align layer (herein RoI Align) is used, which is designed to extract features from proposal boxes from the feature map and regress the classification and bounding box. RoI Align can efficiently decrease the misalignments between the RoI and the extracted feature. Simultaneously, a branch for contour prediction is added to describe the contour of object. The loss function is defined as the sum of the loss of classification, bounding box and contour. When the loss function is converged, the classification and contour of target object are achieved.

2.2. Grasping Network for Pose Estimation

The pose of the target object is estimated in the region of the contour. Consequently, the z-axis of the end-effector is vertical to the object when grasping. RGB-D images are taken along the viewing angle, thus, the distance between the object and the end-effector is equal to the depth information of the image. With the image, the position of the target object (u, v) can be obtained in a pixel coordinate system, while the orientation is expressed as an angle (q) around the geometric center of the contour. By dividing the contour into k-parts around the circle, where each part of the angle is equal to 360/k, then the grasping orientation estimation is turned into an k-way binary classification problem. The region of the contour is input into the grasping network, which uses AlexNet [10] as a backbone. Here, via convolutional layers, the feature map is obtained, and fully-connected layers are used to converge for the classification of grasping orientation. A more detailed description is outlined in our former work [38].

2.3. Steps of Pose Transformation

With instance segmentation and pose estimation, the position in pixel coordinates and the orientation angle are obtained to locate the target object, which can be expressed as (u, v, q). For

manipulator control, this should be transformed by two steps. Firstly, the pose expressed in pixel coordinates is transferred into the camera coordinate, as shown in Equation (1):

$$\begin{cases} z_c \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \\ \boldsymbol{R}_c = \boldsymbol{R}(z,q) \cdot \boldsymbol{R}(y,0) \cdot \boldsymbol{R}(x,0) \end{cases} \tag{1}$$

where $\boldsymbol{K}$ expresses the internal parameters matrix of camera, z_c represents the depth information of image, $\boldsymbol{R}(\cdot)$ represents rotation matrix operator, and $\boldsymbol{R}_c$ represents the attitude matrix.

The second step is as follows, where the pose expressed in camera coordinates is transferred into a joint angle sequence of the manipulator, as shown in Equation (2):

$$\begin{cases} \begin{bmatrix} \boldsymbol{R}_c & \boldsymbol{p}_c \\ \boldsymbol{0} & 1 \end{bmatrix} = {}^{c}_{b}\boldsymbol{T} \begin{bmatrix} \boldsymbol{R}_b & \boldsymbol{p}_b \\ \boldsymbol{0} & 1 \end{bmatrix} \\ \theta = f_{inv}\left(\begin{bmatrix} \boldsymbol{R}_b & \boldsymbol{p}_b \\ \boldsymbol{0} & 1 \end{bmatrix} \right) \end{cases} \tag{2}$$

where ${}^{c}_{b}\boldsymbol{T}$ represents the transformation matrix between the camera coordinates and the base coordinates of the manipulator. The position of the target object is expressed as $(u,v)^{\mathrm{T}}$ in pixel coordinates, $\boldsymbol{p}_c = (x_c, y_c, z_c)^{\mathrm{T}}$ in camera coordinates, and $\boldsymbol{p}_b = (x_b, y_b, z_b)^{\mathrm{T}}$ in the base coordinates of the manipulator. $\theta = \{\theta_i\}$, $i = 1, \ldots, 7$ represents the joint angle sequence, and f_{inv} represents the inverse kinematics function of the manipulator.

3. The Method of ISPF for Pose Transformation

The process of the self-recognized grasping operation is presented above. Instance segmentation, pose estimation, and pose transformation, are all indispensable in achieving self-recognition. However, the traditional method of inverse kinematics in Equation (2) is not adaptable. A new method is required to achieve the expression of all feasible solutions, rapid calculation, and optimal solution selection. In this section, a method of ISPF which relies on manipulator configuration simplification is proposed. The configuration of the redundant manipulator is simplified based on the possible locus circle of the elbow node. Then, the relationship of joint angles is derived analytically, and all the feasible solutions can be calculated. By introducing an optimal factor, the corresponding optimal solution can be selected.

3.1. Manipulator Simplification and Parameters Definition

Considering the configuration of a 7-DOF manipulator, it can be simplified into four parts according to three nodes, namely, the wrist, elbow, and shoulder, which are labeled as P_1, P_2, P_3. The node of the base and the node of the end-effector are labeled as P_0, P_T respectively. The simplified configuration is shown in Figure 3. In the figure, the possible position of P_2 forms a circle, which is named as the locus circle of P_2. The locus circle has the following properties:

a. The center of the locus circle is located on the line between nodes P_1 and P_3.
b. The line between P_1 and P_3 is normal to the plane of locus circle.

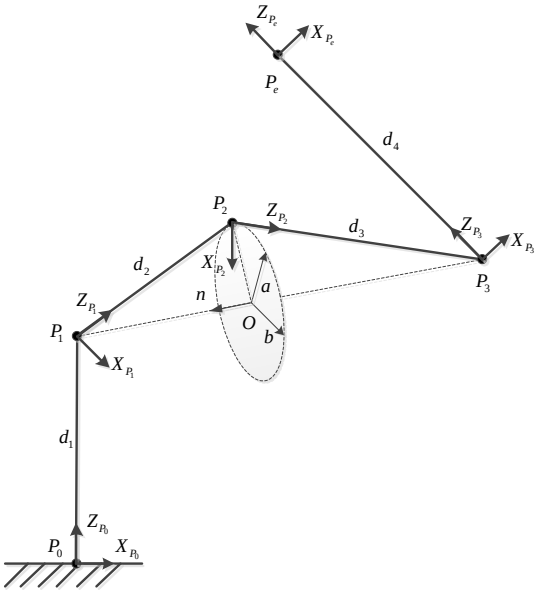


Figure 3. Simplification of the 7-DOF (degrees of freedom) manipulator.

After simplification, the parameters of manipulator can be defined as follows:

$\boldsymbol{P}_e$: The position of the end-effector of the manipulator.
$\boldsymbol{R}_e$: The attitude matrix of the end-effector of the manipulator.
$\sum_m$: The m^{th} joint coordinate system, $m = 0, 1, 2, \ldots, 7, T, P_i$, where $\sum_0$ represents the base coordinate system, $\sum_T$ represents the coordinate system of the end-effector, and $\sum_{P_i}$ represents the coordinate system at node P_i, $i = 0, 1, 2, 3, 4, e$.
${}^{j}_{i}\boldsymbol{T}$: Transformation matrix between the j^{th} and i^{th} coordinate system.
$\boldsymbol{P}_i = (x_i, y_i, z_i)^{\mathrm{T}}$: The position coordinate of the i^{th} node, $i = 0, 1, 2, 3, 4, e$.
$\boldsymbol{o} = (o_1, o_2, o_3)^{\mathrm{T}}$: The center of locus circle of P_2.
$\boldsymbol{a} = (a_1, a_2, a_3)^{\mathrm{T}}$: Arbitrary vector in the plane of locus circle.
$\boldsymbol{b} = (b_1, b_2, b_3)^{\mathrm{T}}$: Vector in the plane of locus circle, fulfilling $\boldsymbol{a} \perp \boldsymbol{b}$.
$\boldsymbol{n} = (n_1, n_2, n_3)^{\mathrm{T}}$: Normal vector of the plane of locus circle.
$d_{1,o}$: The distance between node P_1 and the center of circle O.
$d_{i,j}$: The distance between the i^{th} and j^{th} node.
$\boldsymbol{l}_{i,j}$: The vector between the i^{th} and j^{th} node.

According to the configuration of manipulator shown in Figure 3, the joint angles are divided into three groups, which are respectively derived in the following section.

3.2. Feasible Solutions Expression

Since nodes P_3 and P_e are located on a rigid link, the attitude of the two nodes are consistent. However, the positions of the two nodes are different. According to ${}^{P_3}_{0}\boldsymbol{T}{}^{T}_{P_3}\boldsymbol{T} = {}^{T}_{0}\boldsymbol{T}$, they can be obtained as follows:

$$\begin{bmatrix} \boldsymbol{R}_e & \boldsymbol{P}_3 \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & {}^{e}_{3}\boldsymbol{P} \\ \boldsymbol{0} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_e & \boldsymbol{P}_e \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{3}$$

Then, the position of $\boldsymbol{P}_3$ can be obtained as follows:

$$\boldsymbol{P}_3 = \boldsymbol{P}_e - \boldsymbol{R}_e \cdot {}^{e}_{3}\boldsymbol{P} \tag{4}$$

where ${}^{e}_{3}\boldsymbol{P} = (0, 0, d_4)^{\mathrm{T}}$. Here, we define the locus circle of node $\boldsymbol{P}_2 = (x_2, y_2, z_2)^{\mathrm{T}}$ as follows:

$$\begin{array}{rcl} x_2 & = & o_1 + r{\cdot}a_1 \cos\varphi + r{\cdot}b_1 \sin\varphi \\ y_2 & = & o_2 + r{\cdot}a_2 \cos\varphi + r{\cdot}b_2 \sin\varphi \\ z_2 & = & o_3 + r{\cdot}a_3 \cos\varphi + r{\cdot}b_3 \sin\varphi \end{array} . \tag{5}$$

where r represents the radius of the circle and $\varphi \in (0, 2\pi)$ represents the phase of circle. In order to solve Equation (5), the coordinate of the center, O, should be firstly obtained. Geometrically,

$$d_2^2 - d_{1,o}^2 = d_3^2 - (d_{1,3} - d_{1,o})^2. \tag{6}$$

Now, the distance between node P_1 and P_3 can be obtained as follows:

$$d_{1,3} = \left(x_3^2 + y_3^2 + (z_3 - d_1)^2\right)^{1/2}. \tag{7}$$

Then,

$$d_{1,o} = \frac{d_{1,3}^2 + d_2^2 - d_3^2}{2d_{1,3}}. \tag{8}$$

Since the radius of circle $r = d_{1,o}$, according to Equation (8), the coordinate value of the center is obtained.

$$\boldsymbol{o} = \frac{d_{1,o}{\cdot}\boldsymbol{l}_{1,3}}{d_{1,3}} \tag{9}$$

Meanwhile, $\boldsymbol{a}, \boldsymbol{b}$ can be calculated as follows:

$$\begin{array}{l} \boldsymbol{a} = \frac{\boldsymbol{l}_{1,3} \times (\boldsymbol{l}_{0,1} \times \boldsymbol{l}_{1,3})}{\|\boldsymbol{l}_{1,3} \times (\boldsymbol{l}_{0,1} \times \boldsymbol{l}_{1,3})\|} \\ \boldsymbol{b} = \frac{\boldsymbol{l}_{1,5} \times \boldsymbol{a}}{\|\boldsymbol{l}_{1,5} \times \boldsymbol{a}\|} \end{array} \tag{10}$$

By substituting Equations (9) and (10) into Equation (5), $\boldsymbol{P}_2$ can be calculated. Accordingly, the coordinate values of nodes P_0, P_1, P_2, P_3, P_e are solved, and the joint angles of $\theta_1, \theta_2, \theta_4$ can be expressed as follows:

$$\begin{array}{l} \theta_1 = \arctan(y_2, x_2) \\ \theta_2 = arc\cos\frac{\boldsymbol{l}_{0,1} \cdot \boldsymbol{l}_{1,2}}{|\boldsymbol{l}_{0,1} \cdot \boldsymbol{l}_{1,2}|} \\ \theta_4 = arc\cos\frac{\boldsymbol{l}_{1,2} \cdot \boldsymbol{l}_{2,3}}{|\boldsymbol{l}_{1,2} \cdot \boldsymbol{l}_{2,3}|} \end{array} \tag{11}$$

Here, we define the plane as $\gamma = \{\gamma \big| P_1 \in \gamma, P_2 \in \gamma, P_3 \in \gamma\}$. Since θ_1, θ_2 can be calculated, $\boldsymbol{P}_3$ is determined by θ_3 and the plane γ changes with θ_3. Here, θ_3 can be treated as the dihedral angle between γ and the plane $\gamma' = \gamma(\theta_3 = 0)$. When $\theta_3 = 0$, ${}^{4}_{0}\boldsymbol{T'} = \begin{bmatrix} {}^{4}_{0}\boldsymbol{R'} & {}^{4}_{0}\boldsymbol{P'} \\ \boldsymbol{0} & 1 \end{bmatrix}$ can be obtained according to Equation (11). Here, $\boldsymbol{P}_3{}' = \boldsymbol{P}_3(\theta_3 = 0) = (x_3{}', y_3{}', z_3{}')^{\mathrm{T}}$ can be calculated.

$$\boldsymbol{P}_3{}' = {}^{4}_{0}\boldsymbol{P'} + {}^{4}_{0}\boldsymbol{R'}.{}^{3}_{2}\boldsymbol{P} \tag{12}$$

where ${}^3_2\boldsymbol{P} = (0, 0, d_3)^{\mathrm{T}}$. The normal vector $\boldsymbol{n}_{123}$ of plane γ and normal vector $\boldsymbol{n}_{123}{}'$ of plane γ' can be expressed as follows:

$$\begin{cases} \boldsymbol{n}_{123} = \boldsymbol{l}_{1,2} \times \boldsymbol{l}_{2,3} \\ \boldsymbol{n}_{123}{}' = \boldsymbol{l}_{1,2} \times \boldsymbol{l}_{2,3}{}' \end{cases}. \tag{13}$$

Then, the dihedral angle, namely, θ_3, can be calculated:

$$\theta_3 = \arccos \frac{\boldsymbol{n}_{123} \cdot \boldsymbol{n}_{123}{}'}{|\boldsymbol{n}_{123}||\boldsymbol{n}_{123}{}'|}. \tag{14}$$

After obtaining $\theta_1 \sim \theta_4$, ${}^4_0\boldsymbol{T}$ can be calculated. According to ${}^4_0\boldsymbol{T}{}^5_4\boldsymbol{T}{}^6_5\boldsymbol{T}{}^7_6\boldsymbol{T}{}^T_7\boldsymbol{T} = {}^T_0\boldsymbol{T}$, we define ${}^4_0\boldsymbol{T}^{-1} \cdot {}^T_0\boldsymbol{T} = \boldsymbol{T} \in \boldsymbol{R}^{4\times4}$, where T_{ij} represents the element of i^{th} row and j^{th} column. $\theta_5 \sim \theta_7$ can be calculated as follows:

$$\begin{cases} \theta_5 = a\tan\frac{T_{23}}{T_{13}} \\ \theta_6 = a\cos \quad T_{33} \\ \theta_7 = a\tan\frac{-T_{32}}{T_{31}} \end{cases}. \tag{15}$$

So far, $\theta = \{\theta_i\}$ is derived by Equations (11), (14), and (15), with which all the feasible solutions for a particular grasping pose can be obtained.

3.3. The Optimization of ISPF

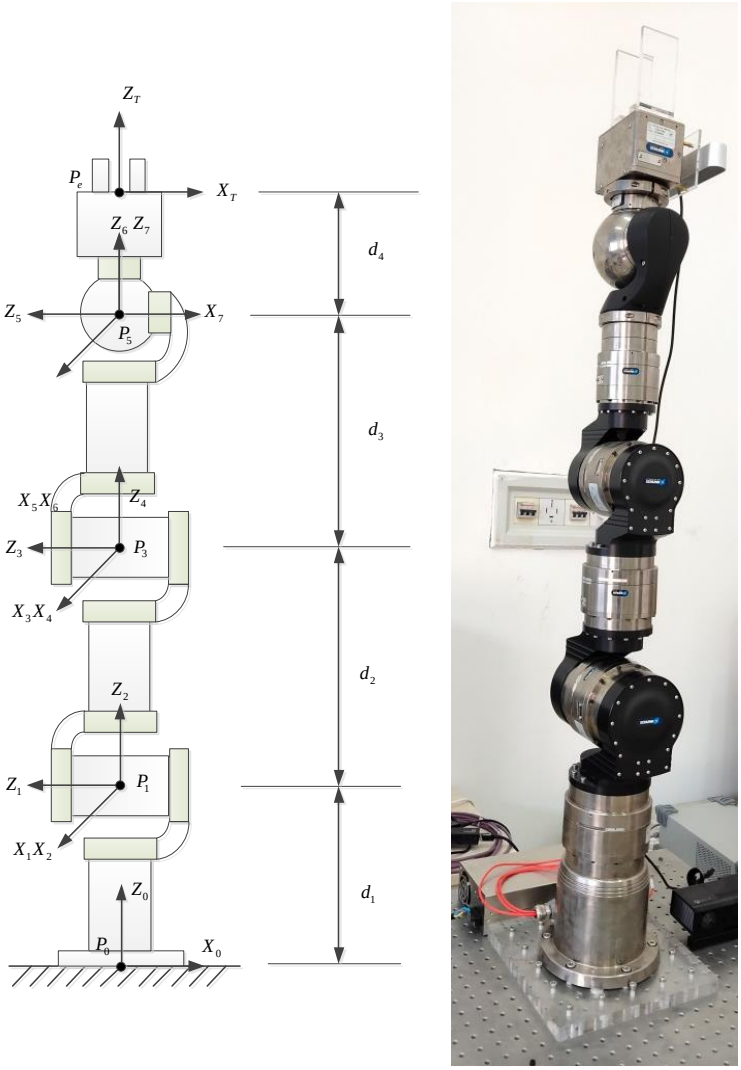
In Section 3.2, the joint angle sequence θ varies with the phase angle φ of the locus circle. In other words, θ can be expressed as the function of phase angle φ of the locus circle:

$$\theta = f_{inv}(\varphi). \tag{16}$$

The relationship between θ and φ is implicitly represented by the function $f_{inv}(\cdot)$. When given the pose of the target object, all feasible solutions in the position form can be calculated. For direct control, the optimal solution should be selected, which can be treated as an optimization problem with a single objective. The optimal solution selection is discussed during grasping experiments in the next section.

4. Simulation and Experiments

The redundant manipulator used in the paper is a 7-DOF serial manipulator made by SCHUNK, whose joint coordinates and configuration are shown in Figure 4. The DH (Denavit-Hartenberg) parameters are listed in Table 1. A ZED camera is set on the end-effector of the manipulator to obtain a RGB-D image. With the manipulator, we used five experiments to validate the proposed methodology and system. Two experiments of instance segmentation and pose estimation based on vision system were carried out to obtain the pose of the target object. Three experiments of pose transformation were carried out to verify the method of ISPF in achieving efficient feasible solution calculation, selecting the optimal solution and showing advantages compared to solutions obtained by the iteration method.



(**a**) (**b**)

Figure 4. (**a**) The joint coordinates of 7-DOF redundant manipulator, (**b**) the configuration of the 7-DOF redundant manipulator.

Table 1. DH (Denavit-Hartenberg) parameters of 7-DOF serial manipulator.

Number	$\alpha(^\circ)$	a(mm)	$\theta(^\circ)$	d(mm)
1	90	0	−90	$d_1 = 380$
2	−90	0	0	0
3	90	0	0	$d_2 = 328$
4	−90	0	0	0
5	90	0	0	$d_3 = 323$
6	−90	0	0	0
7	0	0	90	0
T	0	0	0	$d_4 = 190$

4.1. Experiment on Finding and Locating Target Object

For the self-recognized grasping operation, the primary step is to the distinguish target object from the unstructured environment. Here, we set some fruit on the shelf shown in Figure 5a. There are two kinds of fruit (an apple and orange), and two apples were placed on different layers. We would like the manipulator to distinguish the three objects and grasp the upper apple by itself.

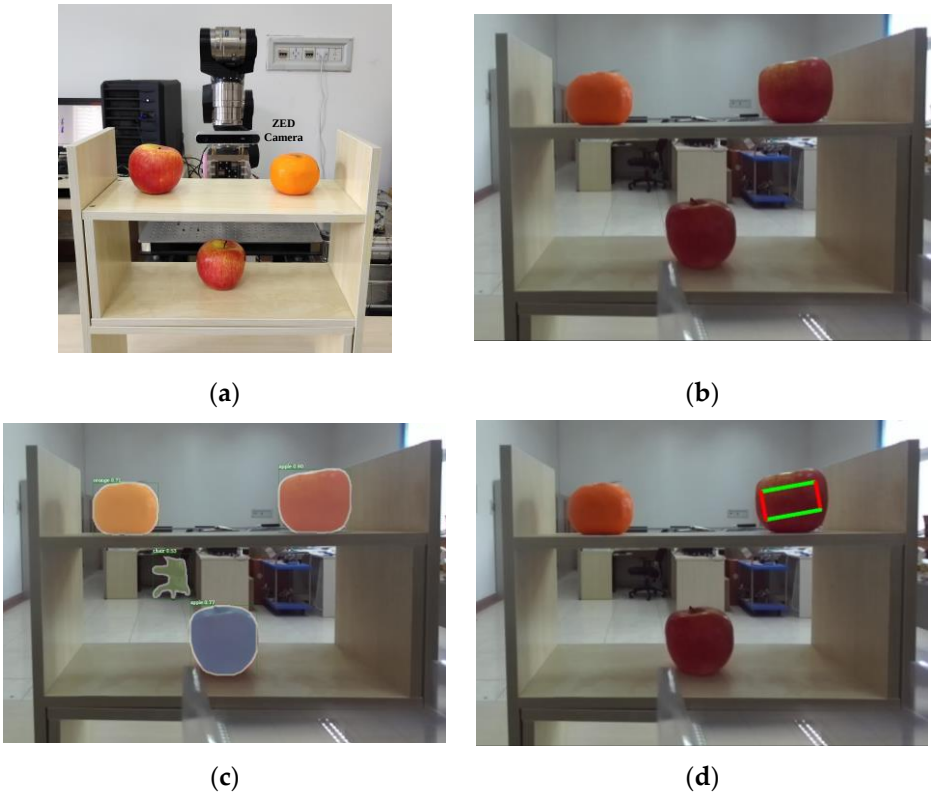


Figure 5. Instance segmentation and pose estimation based on a ZED camera set on the spatial redundant manipulator. (**a**) The experiment scene of grasping, (**b**) image obtained by ZED camera, (**c**) the result of instance segmentation, (**d**) and the result of pose estimation.

The experimental scene obtained via a camera is shown in Figure 5b. The shelf was set as an obstacle and limitation to the workspace of the robot in its grasping operation. Different objects belonging to the same category (apple) were set to confuse the recognition of the target object. The background is a common lab, which is considered as clutter, confusing the feature detection and pose estimation. The experimental scene can be treated as a typical unstructured environment.

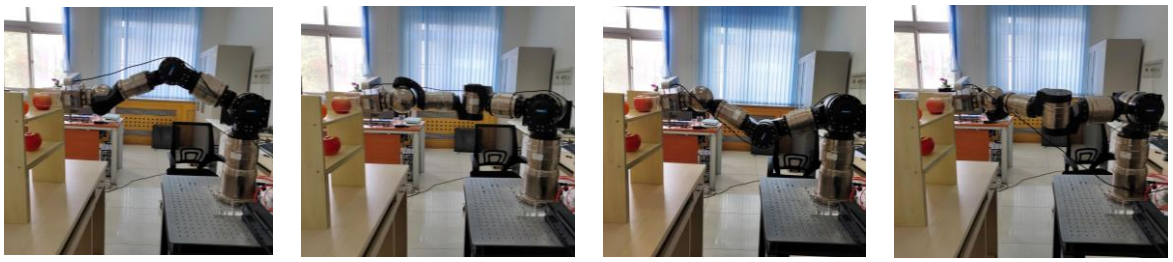
The instance segmentation algorithm is devoted to mark the object and provide a result, as shown in Figure 5c. The fruit on the shelf can be clearly marked by category, and the two apples are also distinguished with different colors. The confidence coefficient of each object is labeled in the figure, and the target apple had the highest confidence coefficient, which was 0.90. Besides, the confused background can be almost eliminated via instance segmentation. As a result, the target apple is clearly recognized for further operation.

After finding the target apple, the position was detected as the center of the contour shown in Figure 5c. The orientation for grasping was treated as a classification problem, based on the grasping network described in Section 2. The contour was divided into 36 parts around the circumference, where each part is equal to a 10° angle, turning the orientation estimation into a 36-way binary classification problem. The orientation estimation result is shown in Figure 5d. The orientation angle of the target apple was classified as 10°.

By taking a photo via the vision system, the manipulator can find the target object from the unstructured environment and obtain the contour of the target object, then obtain the position and orientation angle autonomously. According to the experiment, the efficiency of instance segmentation and grasping network is verified via self-recognized grasping. The benefit from instance segmentation and grasping orientation estimation was used, concerning the contour of the target object. The contour marked by instance segmentation is more accurate than bounding box classification, which greatly reduces the range and time cost for the grasping network to estimate the grasping orientation.

4.2. Feasible Solutions Based on the ISPF Method

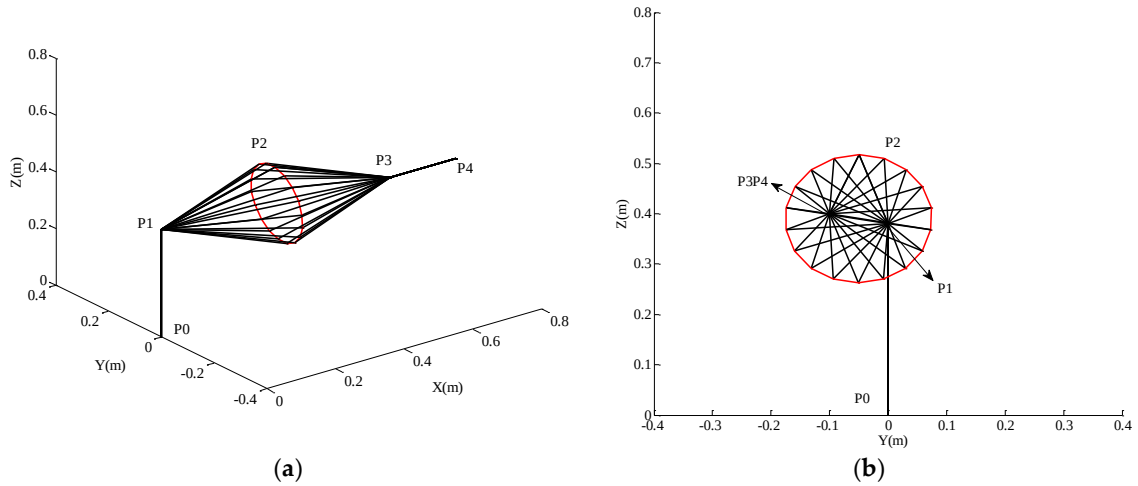
By the experiments in Section 4.1, the position of target object and grasping orientation angle were obtained. The pose of the end-effector, namely, $[0.78\text{m}, -0.10\text{m}, 0.40\text{m}, -1.57\text{rad}, -0.17\text{rad}, -1.57\text{rad}]$, could be obtained according to Equation (1). With the method of ISPF, kinematics and transformation matrices between joint coordinates were calculated by the DH parameters listed in Table 1. Then, solutions for successful grasping could be obtained by changing the value of phase angle φ in $[0, 2\pi]$. In Figure 6, four representative solutions (obtained at $\varphi = 0°$, $\varphi = 90°$, $\varphi = 180°$, and $\varphi = 260°$) are shown to achieve grasping operation.



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

Figure 6. Grasping operation under different phase angles of φ. (**a**) $\varphi = 0°$, (**b**) $\varphi = 90°$, (**c**) $\varphi = 180°$, (**d**) $\varphi = 260°$.

From Figure 6, all four configurations of the redundant manipulator could successfully achieve grasping operation. Different joint angles correspond to different joint positions and manipulator configurations. By varying φ, which varies in $[0, 2\pi]$, at an interval of 20°, 18 groups of inverse solutions were calculated, and the corresponding configurations are shown in Figure 7. Figure 7a is a general view of the solutions, and Figure 7b is viewed from the Y-Z plane. It clearly shows that the possible positions of P_2 form a circle. When φ varies, all the feasible solutions can be obtained and represented, as shown in Figure 7a. This verifies with the method of ISPF, where the manipulator can efficiently calculate the entire feasible solution set. Besides, the shape formed by the possible solutions confirms the validation of manipulator simplification and the locus circle application outlined in Section 2.



(**a**) (**b**)

Figure 7. Simplified manipulator configurations of multiple groups of solution. (**a**) A general view. (**b**) View from Y-Z plane.

4.3. The Optimal Solution with ISPF

During $\varphi \in [0, 2\pi]$, countless solutions exist which can achieve the same pose at the end-effector of the redundant manipulator. However, for grasping operation, only one group of solutions are

needed. In order to find the optimal solution, an optimizing factor can be introduced. Manipulability is an important and common factor representing the dexterity of robots. The optimal solution, when determined by manipulability, has a better adaption for the unstructured environment, especially in terms of humanoid control and dexterous manipulation.

$$\omega = \sqrt{\boldsymbol{J}(\theta)\cdot\boldsymbol{J}(\theta)^{\mathrm{T}}} \tag{17}$$

In Equation (17), $\boldsymbol{J}(\theta)$ represents the Jacobian matrix of the manipulator at joint angles θ, where $\theta = [\theta_1, \theta_2, \ldots, \theta_7]^{\mathrm{T}} \in \boldsymbol{R}^{7\times 1}$. It can be calculated by the DH parameters of the robot and joint angles θ. ω represents the manipulability of manipulator. Since θ has been derived by the ISPF method, corresponding manipulability can be obtained during $\varphi \in [0, 2\pi]$, according to Equation (17). The variation of manipulability related to φ is shown in Figure 8.

From Figure 8, the maximum value of manipulability can be found at $\varphi = 260°$, and the corresponding solution is $\theta_{\max} = [-77.0459\ \ 92.1113\ -99.9610\ \ 46.2262\ \ -0.3588\ -33.1059\ \ 90.4780](°)$. The minimal value of manipulability is at $\varphi = 40°$, where the corresponding solution is $\theta_{\min} = [-114.9950\ \ 70.7922\ \ 42.8671\ \ 46.2262\ -21.7377\ -16.7663\ -19.3553](°)$. The configurations at the minimum and maximum of manipulability are shown in Figure 9. Although the numerical difference of manipulability is not obvious, the configurations have a wide difference at the minimum and maximum of manipulability. Through the introduction of manipulability, the most dexterous grasping configuration can be selected from all the feasible solutions, with which dexterous grasping manipulation can be achieved.

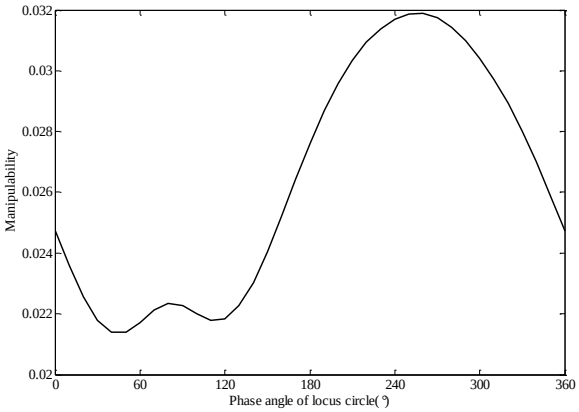
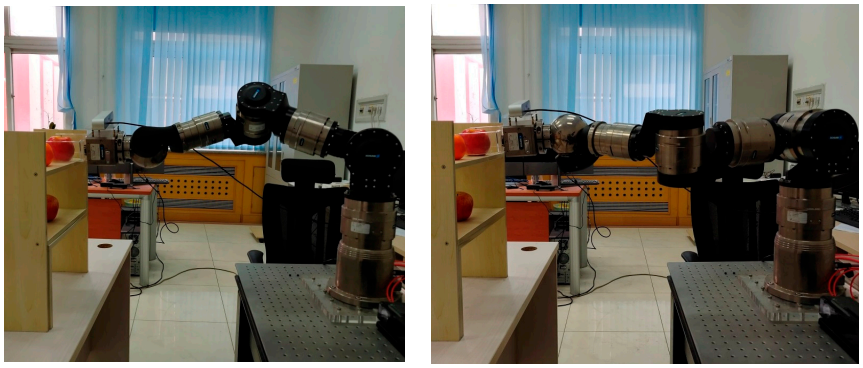


Figure 8. Manipulability related to phase angle φ.



(**a**) (**b**)

Figure 9. The configuration at the minimum and maximum of manipulability. (**a**) The configuration at the minimal of manipulability. (**b**) The configuration at the maximum of manipulability.

Via the introduction of manipulability, manipulation can be expressed as a single object optimization problem of obtaining the optimal solution for self-recognized grasping operation.

$$\begin{aligned} find: &\ \varphi \\ \max: &\ \omega = \sqrt{\boldsymbol{J}(\theta)\cdot\boldsymbol{J}(\theta)^{\mathrm{T}}} \\ &\theta = f_{inv}(\varphi) \end{aligned} \tag{18}$$

With Equation (18), a unique θ value can be obtained to achieve dexterous self-recognized grasping operation with a redundant manipulator. Additionally, other optimal factors, such as the optimal torque, least time cost, and so on, can be introduced to replace manipulability in Equation (18), which will greatly expand the ISPF method in achieving various optimizations for various grasping operation requirements.

4.4. Comparison with the Iteration Method

The iteration method can also achieve pose transformation from the end-effector of the manipulator to joint angles and obtain a group of θ values for grasping operation. However, it requires a proper initial configuration. If the initial configuration is not reasonable, no feasible solution can be reached. When the initial configuration is proper, a group of joint angles can be iterated and converged. In this part, the inverse solutions obtained with iteration method are compared with ones obtained with the ISPF method.

In Table 2, 6 groups of initial configurations are listed to calculated joint angles with the iteration method. For the 1st group, the inverse solution cannot be obtained due to an unreasonable initial configuration. For the other groups, inverse solutions can be converged. The manipulability of these solutions were calculated to compare the solution calculated by the ISPF method, which is shown in Figure 10.

Table 2. Inverse solution with iteration method under different initial configurations.

No.	Initial Configuration (°)	Iteration Method	
		Reachable	Inverse Solution (°)
1	[−20, −20, 0, 0, 0, 0, 0]	No	——
2	[0, 60, 0, 30, 0, 0, 0]	Yes	[−86.68, 69.02, −35.71, 46.48, −19.70, −28.69, 41.19]
3	[10, 10, 10, 10, 10, 10, 10]	Yes	[66.19, −69.73, −141.38, 46.72, −23.00, −17.52, −24.06]
4	[−10, −10, −10, −10, −10, −10, −10]	Yes	[78.50, −65.03, 5.10, −46.85, −25.85, 22.71, −170.49]
5	[−20, 0, 0, 20, −20, 0 − 20]	Yes	[−116.24, 71.07, 45.38, 47.45, −20.48, −17.05, −33.16]
6	[10, 20, 30, 40, 30, 20, 10]	Yes	[−84.77, 70.29, −41.19, 46.68, −18.00, −29.38, 44.80]

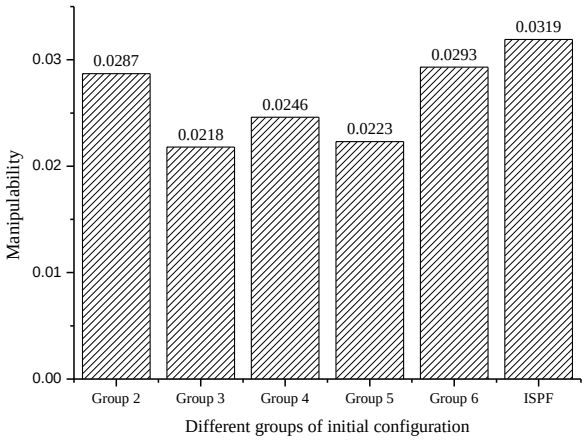


Figure 10. Comparison of manipulability at different groups of initial configurations, obtained by iteration and ISPF methods.

In Figure 10, the manipulability of each solution calculated by the iteration method is smaller than the solution calculated by the ISPF method at $\varphi = 260°$. Although solutions can be obtained by the iteration method, they can hardly be optimal. For self-recognized grasping operation in unstructured environments, various constraints should be guaranteed. The introduction of manipulability makes the manipulator achieve singularity avoidance and joint limit avoidance, leading to better adaption than with the iteration method in unstructured environment applications. Besides, considering the limit of the unstructured environment on the workspace of the manipulator, it is convenient for the ISPF method to adjust feasible solutions achieving obstacle avoidance.

With the experiment, the efficiency of the ISPF method in obtaining the optimal solution for self-recognition grasping operation was verified. Compared with the iteration method, the ISPF method requires no default initial configuration. All the feasible solutions can be calculated, and the optimal one can be selected by introducing an optimal factor. Additionally, it is conveniently programmed with a high computational efficiency, supporting self-recognized grasping, making it a better choice than the iteration method in achieving pose transformation.

5. Discussion and Conclusions

In order to achieve self-recognized grasping operation with a redundant manipulator in an unstructured environment with clutter, the processes of instance segmentation, pose estimation, and pose transformation are described in the present paper. The three processes are indispensable, with which the manipulator can find, locate, and grasp target objects using its vision system, without any outside help. With instance segmentation, different categories of objects can be precisely distinguished with a high confidence coefficient, and the contours of objects can be clearly described. As a result, the range is precisely limited for highly efficient and accurate pose estimation. Via pose estimation with a grasping network, grasping attitude estimation is transferred into a multi-way classification problem, and the attitude represented by the orientation angle is conveniently transferred to a pose in Cartesian coordinates for grasping control. The two processes can work spontaneously once a RGB-D image is obtained by the vision system.

For pose transformation, we propose an inverse solution method at the position form to make it adaptable for self-recognition. The ISPF method requires no default initial configuration and can represent all of the feasible solutions. Besides, by introducing an optimal factor of manipulability, the optimal solution can be selected, achieving humanoid control and dexterous manipulation. Besides, the optimal factor can be replaced according to requirement of optimal control. The ISPF method can be expanded for various optimizations, such as optimal torque, least time cost, and so on. This validates the universality and extendibility of the ISPF method in the optimal control of robots. Further, the characteristic of being conveniently programmed with high computational efficiency makes it quite suitable for self-recognized grasping and real-time manipulator control.

Author Contributions: Conceptualization, T.L.; Data curation, X.S.; Methodology, T.L. and S.Z.; Software, S.Z.; Supervision, C.L.; Visualization, C.W.; Writing—original draft, T.L.; Writing—review and editing, T.L. and X.S.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61802363 and 61774157, Key Research Program of Frontier Sciences, CAS, grant number QYZDY-SSW-JSC037, and the Natural Science Foundation of Beijing, grant number 4182075.

Acknowledgments: The authors would like to thank the State Key Laboratory of Transducer Technology, Institute of Electronics Chinese Academy of Sciences for instrumentation and equipment supports.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bicchi, A.; Kumar, V. Robotic grasping and contact: A review. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 24–28 April 2000; pp. 348–353.
2. Bohg, J.; Morales, A.; Asfour, T.; Kragic, D. Data-Driven Grasp Synthesis—A Survey. *IEEE Trans. Robot.* **2014**, *30*, 289–309. [CrossRef]
3. Jiang, Y.; Moseson, S.; Saxena, A. Efficient grasping from rgbd images: Learning using a new rectangle representation. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3304–3311.
4. Saxena, A.; Driemeyer, J.; Ng, A.Y. Robotic grasping of novel objects using vision. *Int. J. Robot. Res.* **2008**, *27*, 157–173. [CrossRef]
5. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
6. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 404–417.
7. Belongie, S.; Malik, J.; Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 509–522. [CrossRef]
8. Zhang, G.; Chanson, H. Application of local optical flow methods to high-velocity free-surface flows: Validation and application to stepped chutes. *Exp. Therm. Fluid Sci.* **2018**, *90*, 186–199. [CrossRef]
9. Zhang, K.; Zuo, W.; Gu, S.; Zhang, L. Learning deep CNN denoiser prior for image restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 3929–3938.
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
11. Cireşan, D.; Meier, U.; Schmidhuber, J. Multi-column deep neural networks for image classification. *arXiv* **2012**, arXiv:1202.2745.
12. Yang, J.; Yu, K.; Gong, Y.; Huang, T.S. Linear spatial pyramid matching using sparse coding for image classification. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; p. 6.
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
14. Dollár, P.; Appel, R.; Belongie, S.; Perona, P. Fast feature pyramids for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1532–1545. [CrossRef] [PubMed]
15. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
16. Luc, P.; Couprie, C.; Chintala, S.; Verbeek, J. Semantic segmentation using adversarial networks. *arXiv* **2016**, arXiv:1611.08408.
17. Romera-Paredes, B.; Torr, P.H.S. Recurrent instance segmentation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 312–329.
18. Ren, M.; Zemel, R.S. End-to-end instance segmentation with recurrent attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6656–6664.
19. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Simultaneous detection and segmentation. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 297–312.
20. Dai, J.; He, K.; Sun, J. Convolutional feature masking for joint object and stuff segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3992–4000.
21. Dai, J.; He, K.; Sun, J. Instance-aware semantic segmentation via multi-task network cascades. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3150–3158.

22. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
23. Collet, A.; Martinez, M.; Srinivasa, S.S. The MOPED framework: Object recognition and pose estimation for manipulation. *Int. J. Robot. Res.* **2011**, *30*, 1284–1306. [CrossRef]
24. Li, Y.; Wang, G.; Ji, X.; Xiang, Y.; Fox, D. Deepim: Deep iterative matching for 6d pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 683–698.
25. Kim, B.-S.; Xu, S.; Savarese, S. Accurate localization of 3D objects from RGB-D data using segmentation hypotheses. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 3182–3189.
26. Schwarz, M.; Milan, A.; Periyasamy, A.S.; Behnke, S. RGB-D object detection and semantic segmentation for autonomous manipulation in clutter. *Int. J. Robot. Res.* **2018**, *37*, 437–451. [CrossRef]
27. Lenz, I.; Lee, H.; Saxena, A. Deep learning for detecting robotic grasps. *Int. J. Robot. Res.* **2015**, *34*, 705–724. [CrossRef]
28. Goldfeder, C.; Ciocarlie, M.; Dang, H.; Allen, P.K. The Columbia grasp database. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008.
29. Kasper, A.; Xue, Z.; Dillmann, R. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *Int. J. Robot. Res.* **2012**, *31*, 927–934. [CrossRef]
30. Kootstra, G.; Popović, M.; Jørgensen, J.A.; Kragic, D.; Petersen, H.G.; Krüger, N. VisGraB: A benchmark for vision-based grasping. *Paladyn J. Behav. Robot.* **2012**, *3*, 54–62. [CrossRef]
31. Pinto, L.; Gupta, A. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–20 May 2016; pp. 3406–3413.
32. Yu, J.; Weng, K.; Liang, G.; Xie, G. A vision-based robotic grasping system using deep learning for 3D object recognition and pose estimation. In Proceedings of the 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China, 12–14 December 2013; pp. 1175–1180.
33. Levine, S.; Pastor, P.; Krizhevsky, A.; Quillen, D. Learning hand-eye coordination for robotic grasping with large-scale data collection. In Proceedings of the International Symposium on Experimental Robotics, Tokyo, Japan, 3–6 October 2016; pp. 173–184.
34. Manocha, D.; Canny, J.F. Efficient inverse kinematics for general 6R manipulators. *IEEE Trans. Robot. Autom.* **1994**, *10*, 648–657. [CrossRef]
35. Parker, J.K.; Khoogar, A.R.; Goldberg, D.E. Inverse kinematics of redundant robots using genetic algorithms. In Proceedings of the 1989 International Conference on Robotics and Automation, Scottsdale, AZ, USA, 14–19 May 1989; pp. 271–276.
36. Lee, C.; Ziegler, M. Geometric approach in solving inverse kinematics of PUMA robots. *IEEE Trans. Aerosp. Electron. Syst.* **1984**, 695–706. [CrossRef]
37. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
38. Shu, X.; Liu, C.; Li, T.; Wang, C.; Chi, C. A Self-Supervised Learning Manipulator Grasping Approach Based on Instance Segmentation. *IEEE Access* **2018**, *6*, 65055–65064. [CrossRef]