ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

Application of Intelligent Grasping Method based on Machine Vision in Six-Degree-of-freedom Manipulator

Kun Wang[1, 2, a], Guoliang Zhang[1, 2, b]

[1] School of Automation and Information Engineering, Sichuan University of Science & Engineering, Zigong 643000, China

[2] Artificial Intelligence Key Laboratory of Sichuan Province, Zigong 643000, China

[a]2399951900@qq.com, [b]zhgl@sohu.com

Abstract

Aiming at the intelligent application of industrial manipulators, the main research is based on machine vision-based intelligent gripping of manipulators. First, noise reduction is performed based on the point cloud data, and then the grasping pose data set is obtained through the sampling algorithm and combined with the CNN network to achieve pose estimation without pre-building object model. Finally, by building a robotic arm control platform based on ROS, the robotic arm grasping based on the deep learning method and the grasping based on the traditional grasping method are completed, and the two methods are analyzed. The experimental results show that the grasping posture acquired by the deep learning method is more conducive to grasping than the traditional grasping method.

Keywords

Robot; Vision; Point Cloud; Grasp.

1. Introduction

With the introduction of Industry 4.0, in order to adapt to flexible manufacturing, robotic arms are required to have a certain degree of autonomy. The gripping task is the most widely used in industry. As early as 1973, Shiria et al. proposed and built a simple vision-based open-loop control system of a robotic arm[1]. However, the positioning accuracy of the open-loop control system is low, so the grasping success rate is not high. After the introduction of deep learning, the control of the robotic arm has gradually changed from simple geometric drive to data drive. Geometry driving needs to know the model of the target object in advance before planning the robot arm, and then solve the motion trajectory of the robot arm according to methods such as kinematics and geometry, so as to realize the grasping task. The data-driven need to be based on a large amount of data to realize the detection and pose estimation of the target. At present, the difficulty in the application of vision-based robotic grasping lies in the low grasping success rate caused by the large deviation of the pose estimation, and the introduction of deep learning aims to improve the accuracy of the captured poses, thereby Improve the success rate of crawling.

To improve the accuracy of grasping pose estimation, one aspect is to improve the positioning accuracy of the robotic arm. Fu Jinsheng et al.[2]proposed a hand-eye calibration method considering the rotation parameters of the robot's pose; The above article all improve the positioning accuracy of the target by improving the positioning accuracy outside the system, but the system portability of this type of method is low; on the other hand, it is data-driven to improve the positioning accuracy of the target. Tri Wahyu Utomo et al.[3]proposed a robotic grasping pose estimation method based on semantic segmentation in a complex environment. Shinji Kawakura et al. proposed a deep learning

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

system based on fine-tuning using static visual data collected by AI[4]. Jamal Banzi et al.[5] proposed a 3D grasping pose estimation based on convolutional neural network that can be regarded as unsupervised learning. Patten Timothy et al. used deep learning to detect the target object model and compare it with the database model to obtain the target pose to be grasped[6]. The appeal articles all use different types of neural networks to obtain the relative pose of the target, but they are all The relative pose obtained by comparing with the model. When dealing with unknown objects, it is difficult to estimate the relative pose of the target. In addition, the gripping operation of the robotic arm also needs to consider whether it is 2D plane gripping or 6-DOF gripping. Su Jianhua et al. proposed a "cage grabbing" that ignores complex models with grabbing objects. By using a specific grab method, the specific shape of the model is ignored. This method is cleverly conceived and simple. But more is to build a neural network to train and output parameters such as the three-dimensional coordinates of the grasping pose and the opening and closing degree of the grasper. Ian Lenz et al. used the method of 5D representation to capture the pose of a rectangular frame to achieve the estimation of the capture pose of the target[7]. Le Tuan Tang et al.[8] proposed a combination of 2D target recognition with deep neural network and 6D target pose estimation based on feature points Gao Mingyu et al.[9] proposed an autonomous grasping of a robotic arm based on deep learning and particle filtering in nonlinear and non-Gaussian scenarios. Andreas ten Pas et al.[10] proposed an effective grasping pose detection algorithm combining neural network and point cloud. Combined with deep learning, the pose estimation of the target can be achieved, but for objects outside the data set, it is difficult to accurately estimate the specific pose. In response to the above problems, this paper builds a visual robotic arm system: the vision part responsible for collecting image data, the image processing part responsible for processing image data and pose estimation, and the control module responsible for controlling the movement of the robotic arm to achieve a point cloud-based grasping position Pose estimation and 6-DOF intelligent grabbing by robotic arm. After camera calibration, depth map registration and hand-eye calibration, RGB images with depth values in the scene can be obtained through the depth camera. Firstly, the depth map is preprocessed, and the supporting plane is eliminated through filtering, outliers are removed, and regions of interest are searched for. Then use the traditional method to detect the rectangular frame representation of the object to be grasped and the object grasping position selected based on CNN; finally, the robot arm control platform built by combining ROS kinetic and Moveit realizes the control and control of the robot arm. Obstacle avoidance planning. Combining the above three subsystems, it realizes a grasping pose estimation that combines ROS and deep learning without constructing a target model in advance and realizes the corresponding grasping operations.

2. Camera Calibration and Hand-Beye Calibration

The image is distorted due to internal and external factors such as the quality of the camera's lens and the relative position of the installation. Therefore, it is necessary to calibrate the internal and external parameters of the camera, as well as the calculation of the distortion parameters, that is, the camera error correction; before using the vision sensor to estimate the target pose, it is also necessary to obtain the pose of the camera relative to the base coordinates of the robotic arm, so as to achieve the acquisition by the camera. The pose of the target relative to the base coordinates of the robotic arm.

2.1 Camera Calibration

Camera imaging involves four coordinate systems: world coordinate system, camera coordinate system, image coordinate system, and pixel coordinate system. The simplified model of the camera imaging principle is shown in Figure 1.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

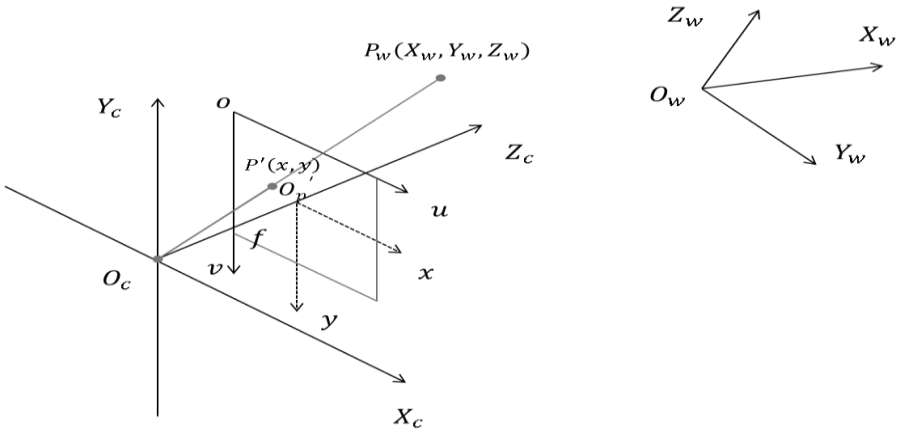


Figure 1. Schematic diagram of camera imaging

Amount them: $O_w - X_w, Y_w, Z_w$: world coordinate system, unit m; $O_c - X_c, Y_c, Z_c$: camera coordinate system, unit m; $O_p^{'} - xy$: image coordinate system, unit mm; $O - uv$: pixel coordinate system, unit pixel; P_w : a point in the world coordinate system; $P^{'}(x, y)$: P_w corresponding points in the camera image; f :focal length, $f = \left\| O_p^{'} - O_c \right\|$.

Through the coordinate conversion from the world coordinate system to the camera coordinate system, the coordinates of a point in the world coordinate system in the camera coordinate system are obtained. Assuming that a point $P_w : [X_w \quad Y_w \quad Z_w]^T$ in the world coordinate system is converted to the point $P_c : [X_c \quad Y_c \quad Z_c]^T$ in camera coordinate system through rotation and translation, The conversion relationship can be represented by a three-dimensional rotation matrix R and a one-dimensional vector T for homogeneous transformation.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{1}$$

The transformation relationship from the camera coordinate system to the image coordinate system: the camera optical centeris O_c used as the origin, the front of the camera is the Z axis to establish the coordinate system, and the mapping point of the camera optical center on the image is the origin of the image coordinate system to establish the image coordinate system. Assuming it is a point P_p in the image coordinate system, in order to facilitate modeling and calculation, flipping the image, marked $P_p^{'}$ as the mapping point P_p, and the relationship between the image points $P_p^{'} : [x \quad y]^T$ and the point $P_c : [X_c \quad Y_c \quad Z_c]^T$ in the camera coordinate system can be calculated by similar triangles:

$$x = f\frac{X_c}{Z_c}, \mathrm{y} = f\frac{Y_c}{Z_c} \tag{2}$$

Write it as a homogeneous matrix as:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

$$Z_c\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}=\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{3}$$

The transformation relationship between the image coordinate system and the pixel coordinate system: because the origin of the image coordinate system is in the image, and the origin of the pixel coordinate is generally the upper left corner of the image, and the measurement unit of the pixel coordinate is the pixel, so the mapping between the two is required Expressed by differential equations. Assuming that the origin in the image coordinate system is $O_p^{'}$, the image point is $P_p^{'}:[x \quad y]^T$, the origin coordinate in the pixel coordinate system is $O:[u_0 \quad v_0]^T$, and the pixel point coordinate is $P_p^{'}:[x \quad y]^T$. Because the pixel is not necessarily a square, the metric of the image coordinate system is used to $dx,\ dy$ represent the physical in the pixel coordinate system. For the mapping relationship of the metric, the following formula can be obtained:

$$\begin{aligned} (u-u_0)dx &= x \\ (v-v_0)dy &= y \end{aligned} \tag{4}$$

Write formula (3) as a homogeneous matrix as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}=\begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

Simultaneously simplify the above equations (1), (2), and (4) to obtain the mapping relationship from the world coordinate system to the pixel coordinate system:

$$Z_c\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}=\begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} R_{3\times3} & T_{3\times1} \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}=M_1M_2\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{6}$$

$M_1\ M_2$ are the internal and external parameter matrices of the camera.

Through the calculation of the camera's internal and external parameter matrices, the camera can be calibrated to correct errors caused by image distortion, thereby improving the accuracy of algorithms such as 3D reconstruction.

Image calibration is to calculate the parameters of radial distortion and tangential distortion, that is to calculate the radial distortion parameter k_1, k_2, k_3 in equation (6), and the tangential distortion parameter p_1, p_2.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

$$\begin{bmatrix} x^{'} \\ y^{'} \end{bmatrix} = \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2p_1 xy + p_2\left(r^2 + 2x^2\right) \\ 2p_2 xy + p_1\left(r^2 + 2y^2\right) \end{bmatrix} \tag{7}$$

x, y is the coordinate of the image point after distortion; $x^{'}, y^{'}$ is the image point coordinates after distortion correction; r is the radius of curvature.

The principle of the depth camera is similar. The depth map imaging principle of the depth camera is based on the light encoding of the scene by obtaining the light source, and then it is obtained by calculation. For example, the Kinect V2 camera uses the ToF (Time of Flight) technology, which uses the camera's photosensitive device to receive the reflected infrared light continuously emitted to the target panel through the infrared light source to calculate the time difference between the light emission and the reception to calculate the distance. In this way, the RGB image and the depth image of the object in the scene to be captured can be obtained.

After obtaining the depth information of the objects in the scene, it needs to be matched with the obtained RGB image, that is, the mapping relationship between the infrared camera and the RGB camera is obtained, so as to obtain accurate point cloud data. Prepare for the crawl experiment.

2.2 Registration of Depth Image and RGB Image

After the internal and external parameters of the camera are calibrated, accurate RGB images and depth maps can be obtained. However, in machine vision, the depth maps generated by the camera often need to be registered to obtain the registered color depth images. Its purpose is to overlap the acquired depth image with the RGB image, that is, to convert the coordinate system of the depth map to the RGB image coordinate system, that is, to calculate $W^{'}$ in formula (8).

$$\begin{bmatrix} u_I \\ v_I \\ 1 \end{bmatrix} = W^{'} \begin{bmatrix} u_{rgb} \\ v_{rgb} \\ 1 \end{bmatrix} \tag{8}$$

Combining equation (2) and normalizing the homogeneous transformation to obtain the transformation from the image coordinate system to the infrared camera:

$$\begin{bmatrix} x_I \\ y_I \\ z_I \\ 1 \end{bmatrix} = z_I \begin{bmatrix} \frac{f}{dx} & 0 & u_0 & 0 \\ 0 & \frac{f}{dy} & v_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_I \\ v_I \\ 1 \\ 1/z_I \end{bmatrix} \tag{9}$$

In the same way, the homogeneous transformation matrix of the image coordinate system to the RGB camera can be obtained, and then the infrared camera can be transformed to the RGB camera uniformly:

$$\begin{bmatrix} x_I \\ y_I \\ z_I \\ 1 \end{bmatrix} = M \begin{bmatrix} x_{rgb} \\ y_{rgb} \\ z_{rgb} \\ 1 \end{bmatrix} \tag{10}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

u_I, v_I, z_I indicates the coordinate system of the depth map and the depth value in the camera coordinate system; $u_{rgb}, v_{rgb}, z_{rgb}$ represents the coordinates of the RGB image and the depth value under the camera coordinates; M is 4×4 matrix; $z_{rgb} \approx z_I$;

According to formula (9) and (10):

$$u_{rgb} = r_{11}u_I + r_{12}u_I + r_{13} + r_{14} \cdot 1/z_I \tag{11}$$

$$v_{rgb} = r_{21}u_I + r_{22}u_I + r_{23} + r_{24} \cdot 1/z_I \tag{12}$$

That is, the coordinate conversion from the depth image to the RGB image is obtained.

2.3 Hand-Eye Calibration

After calculating the mapping relationship from the world coordinate system to the pixel coordinate system, it is also necessary to determine the mapping relationship between the pose of the object in the scene and the end of the robot arm through hand-eye calibration. There are two main ways to install vision sensors and robotic arms: "eyes on hands" and "eyes outside of hands". However, hand-eye calibration is required to determine the coordinate transformation relationship between the camera coordinate system and the end gripping jaws of the robot arm and the object to be grasped. Take "eyes outside of hands" as an example:

Fix the calibration board to the end of the robotic arm, and install the camera and the base of the robotic arm to build the hand-eye calibration system. The homogeneous transformation matrix of the camera coordinate system in the base coordinate system of the robot arm is D: the homogeneous transformation matrix of the end of the robot arm in the base coordinate system is A, The homogeneous transformation matrix in the coordinate system is X, the relational expression can be obtained: $D = AXB$ that is B, the hand-eye calibration is to obtain the transformation matrix D.

Through the positive kinematics modeling of the robotic arm, the mapping relationship between the end of the robotic arm and the base coordinate system of the robotic arm can be calculated; through the camera calibration, the world coordinate system (here the index fixes the board pose) and the camera coordinate system can be calculated Therefore, the transformation relationship between the phase attitude and the base coordinate system can be obtained by calculating the transformation matrix, so that the mapping relationship between the end of the robot arm and the object to be grasped can be calculated by the camera imaging principle, and the positioning of the object can be realized. In order to obtain the transformation matrix X, AR tags, high-precision instrument measurement, system identification and other methods can be used to calibrate the solution matrix X. The following takes the AR tag method with low cost and high calculation accuracy as an example for hand-eye calibration. By using the AR tag of the specified size to be fixed at the end of the robotic arm, the command or teaching mechanism is used to control the movement of the robotic arm tag in the camera's field of view, and the relative movement data of the tag and the camera are recorded and the pose relationship between the end of the robotic arm is formed Mapping relationship, and then obtain the transformation relationship between the camera and the end of the robotic arm, that is, the transformation matrix X.

By calibrating the camera's internal and external parameters, depth map and RGB image registration, and hand-eye calibration, the relationship between the object in the robotic arm workspace and the two-finger gripper at the end of the robotic arm can be obtained, and the object to be grasped can be determined the trajectory of the robotic arm can be planned by solving the inverse kinematics.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

3. Grabbing Pose Estimation

Vision-based robotic arm grasping has always been a hot topic in the field of robotics research and it is also a research difficulty. The difficulty is that the grasping pose is difficult to accurately estimate, resulting in a low grasping success rate. The traditional grasping method is to locate the target object by acquiring various sensor data or through teaching methods to achieve grasping, without the need to estimate the posture of the target object. Although there is no need to build a CAD model of the target in advance, but in a complex environment, the acquired data has large errors. Using such noisy data as input will reduce the success rate of capture and make it difficult to apply Into real applications. In order to improve the success rate of grasping, the focus is to achieve the accuracy and precision of target pose estimation and grasping attitude estimation. Based on improving the accuracy of the target pose estimation, the positioning accuracy of the camera can be improved. Target pose recognition in complex environments uses more methods to obtain grasped poses through deep learning.

3.1 Implementation of Traditional Grasping Methods

Based on the traditional grasping method, it is mainly to extract the target from the scene, so as to locate the target. However, to locate the original point cloud data, there may be a lot of noise, and the point cloud data is complicated and the calculation speed is slow. Therefore, some preprocessing of the original point cloud data is required. For example, when the field of view of the camera is large and the range of the target is relatively small, you can use straight-pass filtering to reduce the detection area of the point cloud image, and select the point cloud data of the area of interest; when the original point cloud density is high, voxels can be used Filtering, using voxel grid sampling to reduce point cloud data and speed up the calculation; when the scene has a supporting plane, random sampling can be used to determine the supporting plane using the principle of three points and one plane, thereby eliminating the point cloud data of the supporting plane. When there is clear noise in the point cloud data, the calculation of the first and second moments of the object will cause large errors. At this time, statistical filtering can be used to remove the outliers of the point cloud image to reduce the error; then after expansion and The corrosion is connected into a closed contour, and the largest contour is selected as the detected target. By calculating the first-order and second-order moments of the contour, the center of gravity and the minimum circumscribed rectangle of the target are obtained, so as to obtain the grasping point and the grasping pose of the target.

3.2 Grabbing based on Deep Learning

Grabbing based on deep learning mainly relies on a data set with a grasping pose or through a training data set to construct a successful grasping method to obtain the grasped pose. For example, a robotic arm grasping pose estimation based on CNN deep learning can train the data set by using the position and deflection angle of the target rectangle to be grasped as the output of the neural network, so as to obtain the correct grasping posture.

Considering the large noise of the original point cloud data, the point cloud is preprocessed first, and after removing the noise, through random and uniform sampling, the feasible grasping posture is selected to form a grasping posture data set, and the set of successful grasping Discriminant function, training data set, get the grasping posture with the highest grasping success rate. Grasp based on the Grasp Pose Detection algorithm (Grasp Pose Detection) method is different from the traditional method of grabbing. It does not need to detect the target in advance but directly extracts the local grab surface data from the sensor data, and merges multiple the grasping pose can be obtained by the method of viewpoint, a depth map of a single angle of view, and the normal vector of the target surface. On this basis, the problem can be described as: given the point cloud data and the geometric description of the manipulator, the grasping posture detection is to identify the configuration of the grasping posture when the gripper is closed. The complete grasping can be described as: applying a certain force to the closed posture configuration of the grasper, and the frictional force formed by this force on the surface of the target must meet the grasping requirements. The GPD algorithm takes the

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

point cloud, the region of interest, the two-finger gripper, and the number of samples N as input, and the 6D grasping pose candidates as the output. The general process is as follows:

1) The point cloud data obtained by the sensor is compressed and denoised through voxel filtering, statistical filtering to remove outliers, etc.;

2) Use straight-through filtering to obtain the ROI (Region of Interest) mark R;

3) Collect N grabbing candidates from the ROI area, and each candidate is a 6D grabber posture;

4) Encode the collected candidates to form a data set;

5) Use a four-layer convolutional neural network (CNN) to evaluate the sample;

6) Filter out the optimal grasping pose according to the evaluation results.

Step 1) De-noise the original point cloud data to obtain the point cloud of the target. Step 2) Filter out the target point cloud data in the region of interest, and rasterize the closed interval voxels into voxels grid; According to the results obtained in step 2), N points are uniformly collected on the surface of the object, and the eigenvector of the matrix is calculated for each sampling point to calculate the local reference system of the sampling point. As shown in formula (13), the orthogonal reference coordinate system is obtained by calculating the eigenvalues of the sampling points and the corresponding eigenvectors, thereby determining a set of grasping postures , and using the local grid search method to translate or rotate the orthogonal reference coordinate system of the sampling points to eliminate the two-finger grasp the non-contact posture of the hand and the point cloud and the posture of the target point cloud in the closed interval of the gripper. After continuous sampling, a feasible grasping posture set is obtained; then the grasping posture data set is passed through the projection body through equation (14) Under the orthogonal reference coordinate system of the hand axis of the gripper, the multi-view coding is then imported into the multi-channel CNN network for evaluation, and each grasping posture is estimated to evaluate the quality of the grasping effect, and finally it is solved. And the necessary factors for grabbing to screen out the correct grabbing posture.

$$M(p)=\sum_{q\in R(C^*)} n(p)n(p)^T \tag{13}$$

$$\begin{cases} I_X(x,y)=\dfrac{\sum_{z\in[1,M]} zV(x,y,z)}{\sum_{z\in[1,M]} V(x,y,z)} \\ I_Y(x,y)=\dfrac{\sum_{z\in[1,M]} zU(x,y,z)}{\sum_{z\in[1,M]} U(x,y,z)} \\ I_n(x,y)=\dfrac{\sum_{z\in[1,M]} \hat{n}(x,y,z)V(x,y,z)}{\sum_{z\in[1,M]} V(x,y,z)} \end{cases} \tag{14}$$

Among them, $\hat{n}(p)$ represents the normal vector of the point ; $V(x,y,z)$ indicates whether the corresponding voxel of the rasterized interval has a value, {0,1}; $U(x,y,z)$ indicates whether the corresponding voxel of the rasterized interval has been detected, {0,1}; $I_X(x,y)$, $I_Y(x,y)$ represent the size of $M\times M$ the image projected on the reference coordinate system; $I_n(x,y)$ represents the three-dimensional decomposition of the normal vector of the point cloud, the size of $M\times M\times 3$;

The four-layer convolutional neural network shown in Figure 2 is used to classify and evaluate the grasping pose data set. The training neural network uses the BigBird data set, which has 125 objects, and each sample has more than 50,000 grab samples with real labels. The data set is randomly sampled to train CNN: 30,000 randomly selected each time sample. Set the stochastic gradient

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

descent method with a learning rate of 0.00025 to train the data set H , add the softmax function before the output layer, and map the output result to $(0,1)$ to estimate the quality of the grasping posture. Finally, the optimal grasping posture is obtained by the solution.

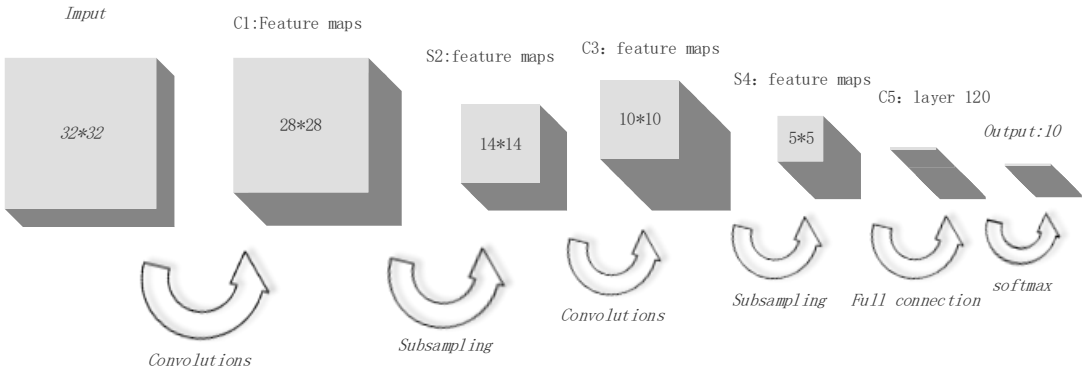


Figure 2. CNN structure diagram

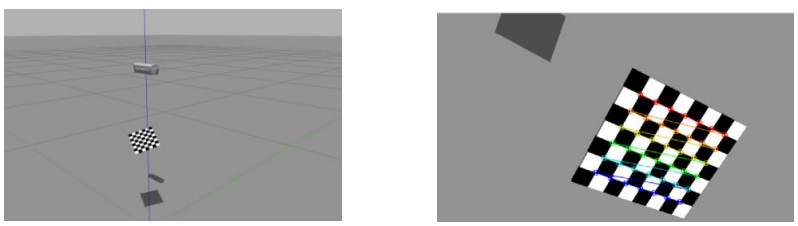
4. Experimental Platform Construction and Simulation Experiment

Build a simulation platform based on ububtu 16.04 and ROS-Kinetic version, use IKfast kinematics parser and Moveit module to perform camera calibration, hand-eye calibration, pose estimation to be grabbed, and grab simulation experiment.

4.1 Camera Calibration Experiment

In order to obtain accurate image data, it is necessary to obtain the internal and external parameters of the camera. Using Zhang Zhengyou's chessboard method to solve the internal and external parameter matrix of the depth camera RealSense D435i. First of all, the camera needs to be fixed to avoid inaccurate image data collected, resulting in large calculation errors. Then move the calibration board continuously, collect at least three RGB images and infrared images through the camera to calculate the internal and external parameter matrix of the camera.

The camera calibration experiment was carried out in the GAZEBO environment in the ROS system. After importing the model of the camera and the calibration plate, 15 moving and complete images of the calibration plate collected by the camera are saved, which is convenient for obtaining the internal and external parameter matrix of the camera later. The following figure shows the experiment based on camera calibration (a) and the collected calibration board image (b).



(a) (b)

Figure 3. Camera calibration experiment diagram

Through the camera calibration function package of the ROS system, the 15 images saved are used to calculate the internal parameter matrix of the calibration. The results of the internal parameter matrix and the distortion coefficient matrix of the RGB camera and the infrared camera are as follow:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

RGB camera internal parameter matrix:

$$\begin{bmatrix} 1179.87560421 & 0 & 638.03605802 \\ 0 & 1179.87131121 & 358.81120248 \\ 0 & 0 & 1 \end{bmatrix}$$

Calculated the RGB camera distortion coefficient matrix by formula (6):

$$\begin{bmatrix} -0.00013445 & 0.00444663 & -0.00020416 & -0.0003812 & -0.00861818 \end{bmatrix}$$

Infrared camera internal parameter matrix:

$$\begin{bmatrix} 1179.30370274 & 0 & 638.89301117 \\ 0 & 1179.32949768 & 362.36037727 \\ 0 & 0 & 1 \end{bmatrix}$$

The distortion coefficient matrix of the infrared camera is calculated by formula (6):

$$\begin{bmatrix} 0.00070351 & -0.00711796 & 0.00054326 & -0.00029897 & 0.00325639 \end{bmatrix}$$

4.2 Depth Map and RGB Image Registration Experiment

After the camera calibration task is completed, accurate RGB image and depth map can be obtained, but the corresponding relationship between the two needs to be established to obtain RGBD image, for which depth map registration is required. Calculate the projection relationship between the acquired set of RGB images and depth images in the camera coordinate system to obtain the transformation relationship from the depth map coordinate system to the RGB image coordinate system, because it can be seen from equation (10), The transformation relationship between the two only involves the first two rows of the transformation matrix. The result of the first two rows of the transformation matrix is as follows:

$$\begin{bmatrix} 1.000132 & -0.001627 & 0.413372 & -17.533134 \\ -0.000119 & 0.999585 & 0.046129 & 0.121611 \end{bmatrix}$$

4.3 Hand-Eye Calibration Experiment

Through the hand-eye calibration experiment of the robot arm and the camera, the relationship between the camera coordinate system and the end coordinate system of the robot arm is obtained. With the preparation of the transformation matrix obtained by the camera calibration and the robot arm D-H modeling method, the mapping relationship between the target object and the robot arm base coordinate system is calculated. The following is mainly combined with the hand-eye calibration function package of the ROS system to achieve hand-eye calibration.

First, fix the camera on the appropriate position of the robotic arm, and adopt the "eye on hand" model. Here we choose a fixed installation at the end of the robot arm at a height of 5cm, and then use the AR tag in the hand-eye calibration function package to achieve hand-eye calibration. Use the calibration label with the size of 20cm and the ID of 582 in the Aruco function package. After loading the model, run the hand-eye calibration package, and the results are as follows:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

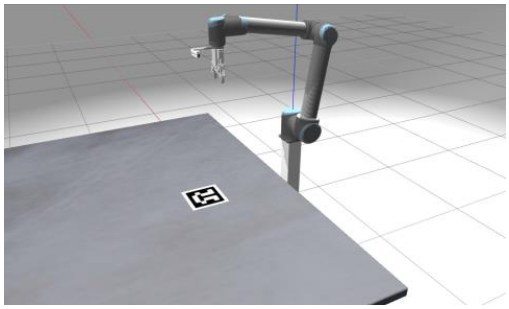


Figure 4. Experimental diagram of hand-eye calibration

According to the instructions of the hand-eye calibration package, keep moving the end of the robotic arm to collect the feature points in the AR tag. After moving 15 positions, the calculated transformation matrix is:

$$\begin{bmatrix} 1.0000 & 0.0012 & -0.0014 & -0.0172 \\ -0.0012 & 1.0000 & 0.0054 & 0.1290 \\ 0.0014 & -0.0054 & 1.0000 & 0.1463 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation matrix is converted into a four-element and position relationship representation, and the result of comparison with the model calculation value is shown in Table 1 below. It can be seen that the error of hand-eye calibration is small.

Table 1. Hand-eye calibration error table

Translation	Measurements(m)	Value(m)	Error(mm)
x	-0.0171515439439	-0.0175	0.35
y	0.129039200607	0.125	4.04
z	0.146263405556	0.1425	3.8
Rotation	Measurements	Value	Error
x	0.999995982804	1	0.01
y	0. 00268604823595	0	2.69
z	0.000687382040816	0	0.69
w	0.000589089946183	0	0.59

4.4 Grabbing Experiment based on GPD Method

By building a robot arm control system based on the ROS-kinetic version, combined with Moveit to realize the planning control of the robot arm in Cartesian space. By preparing the experiment, the transformation matrix from the target to the base coordinates of the robot arm can be obtained to realize the positioning of the target. By loading the GPD algorithm, the grasping pose estimation of the target is realized, and then the robot arm is controlled by Moveit to realize the trajectory planning.

First realize the crawling experiment based on the traditional method. Traditional visual-based grasping needs to select the grasping point of the target object and construct a rectangular frame to achieve grasping. First, an assumption needs to be made: the texture of the object is uniform. According to the results of the pre-experiment, an accurate point cloud image is obtained, and then

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

some processing is performed on the image: image segmentation; target contour extraction; feature points based on torque to describe the object's centroid and direction; select gripping points along the centroid ; Finally, an evaluation is made based on the grab points. For the above steps, a simulation experiment was carried out on the UR10 model and the two-finger gripper; the target model block, banana, and Coke were loaded on the desktop. Based on the experimental platform built, the pose estimation and grasping experiments of the target are carried out. Fig. 5 below shows the grasping pose experiment based on the description of the target rectangle.

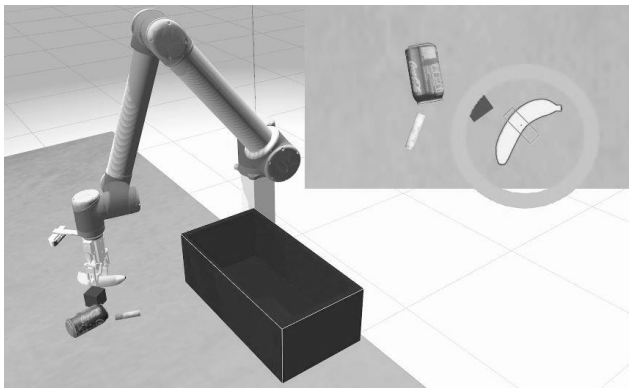


Figure 5. Traditional crawling experiment diagram

Grasping experiments based on the GPD algorithm are mainly integrated into the convolutional neural network in the target pose estimation experiment. The pose estimation based on the GPD algorithm no longer describes the pose grasped by the grasper in the form of a rectangular box, but generates a data set of samples grasped by the grasper through random and uniform sampling, and then filters and evaluates the grasped by a neural network Pose, and finally select the grabbing pose with the highest evaluation as the output to achieve the grabbing of the target. Fig. 6 below is a grabbing experiment based on the GPD algorithm.

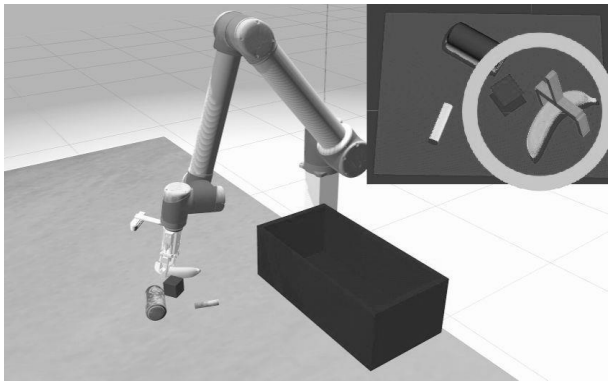


Figure 6. Grabbing experiment diagram based on deep learning

5. Conclusion

In this paper, the pose estimation and grasping experiment of the object to be grasped are realized based on vision. First, the camera's internal and external parameter matrix is obtained by camera calibration to obtain the distortion coefficient to correct the distortion of the image; the RGBD image is obtained by the depth map registration; through the hand-eye calibration experiment and the camera's internal parameter matrix, the transformation between the target point and the end of the robot arm is calculated relation. Then, the traditional rectangular frame method is used to describe the grasping pose of the target and the grasping experiment is carried out. Finally, based on the GPD method, the pose estimation and grasping experiment of the target are realized. Summarize two

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0054

methods: The calculation based on the traditional mechanical arm grasping is simple, but it is difficult to obtain the accurate grasping pose of the target in a complex environment, and the environmental noise has a great influence on the grasping pose estimation. The mechanical arm grasping based on the GPD method can filter out a better grasping posture through the neural network, thereby greatly improving the grasping success rate.

References

[1] Shirai Y,Inoue H.Guiding a robot by visual feedback in assembling tasks[J].Pattern Recognition, 1973, 5(02):99-108.

[2] Fu Jinsheng,Ding Yabin,Huang,et al. Hand-eye calibration method with a three-dimensional-vision sensor considering the rotation parameters of the robot pose[J]. International Journal of Advanced Robotic Systems,2020,17(6).

[3] Tri Wahyu Utomo,Adha Imam Cahyadi,Igi Ardiyanto. Suction-based grasp point estimation in cluttered environment for robotic manipulator using deep learning-based affordance map[J]. International Journal of Automation and Computing,2021(prepublish).

[4] Shinji Kawakura,Ryosuke Shibasaki. Distinction of edible and inedible harvests using a fine-tuning-based deep learning system[J]. Journal of Advanced Agricultural Technologies,2019,6(4).

[5] Jamal Banzi,Isack Bulugu,Zhongfu Ye. Learning a Deep Predictive Coding Network for a Semi-Supervised 3D-Hand Pose Estimation[J].IEEE/CAA Journal of Automatica Sinica, 2020, 7(05): 1371-1379.

[6] Patten Timothy,Park Kiru,Vincze Markus. DGCM-Net: dense geometrical correspondence matching network for incremental experience-based robotic grasping[J]. Frontiers in Robotics and AI,2020.

[7] Ian Lenz,Honglak Lee,Ashutosh Saxena. Deep learning for detecting robotic grasps[J]. The International Journal of Robotics Research,2015,34(4-5).

[8] Le Tuan Tang,Le Trung Son,Chen Yu Ru,et al. 6D pose estimation with combined deep learning and 3D vision techniques for a fast and accurate object grasping[J]. Robotics and Autonomous Systems, 2021 (prepublish).

[9] Gao Mingyu,Cai Qinyu,Zheng Bowen,et a1. A hybrid YOLOv4 and particle filter based robotic arm grabbing system in nonlinear and non-Gaussian environment[J]. Electronics,2021,10(10).

[10] Andreas ten Pas,Marcus Gualtieri,Kate Saenko,et al. Grasp pose detection in point clouds[J]. The International Journal of Robotics Research,2017,36(13-14).