Distance Perception of Unknown Environment Based on Kinect

Yao liu[1, a], and Yangwei Cheng[2, b]

[1]Key Laboratory of Metallurgical Equipment and Control Technology, Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China.

[2]Hubei Key Laboratory of Metallurgical Transmission and Manufacturing Engineer, Wuhan University of Science and Technology, Wuhan 430081, China.

[a]540916774@qq.com, [b]970619543@qq.com

Keywords: Kinect; Manipulator; Distance perception

Abstract: In order to solve the problem of manipulator working in unknown environment, an accurate sensing method based on Kinect sensor is proposed. In the process of working, the manipulator first establishes the coordinate system of Kinect and the manipulator, and constructs the Bursha coordinate conversion model; Then the model is solved by using the linear total least squares algorithm. Finally, according to the grab point coordinate information obtained by Kinect, it is converted from the coordinate conversion to the manipulator coordinate system; In order to achieve the robot's position on the target object. According to the target positioning system provided, the manipulator realizes the precise perception of different objects.

1. Introduction

With the development of network skills, the network operation of the manipulator is also the future development direction. Industrial robot is a kind of high-tech automatic production equipment developed in recent decades.[1] Industrial robots are an important branch of industrial robots. Its feature is that it can be programmed to complete various expected tasks. It has the advantages of both human and machine in terms of structure and performance, and it especially embodies human intelligence and adaptability. The accuracy of manipulator operations and the ability to complete operations in various environments have broad prospects for development in all areas of the national economy.

2. Kinect introduction

Kinect, as a 3D depth sensor device, has many applications in many fields such as machine vision, computer graphics, virtual reality, and reverse engineering. Kinect is simple, powerful, and inexpensive. [2] Its appearance is shown in the following figure, including random infrared point cloud projectors, infrared cameras, color cameras, and microphone arrays.

Gets RGB color stream and target depth stream. A series of initial tasks are required before image processing, including Kinect sensor initialization, RGB color flow initialization, and depth flow initialization.[3] The data initialization flowchart is shown in Figure 1.

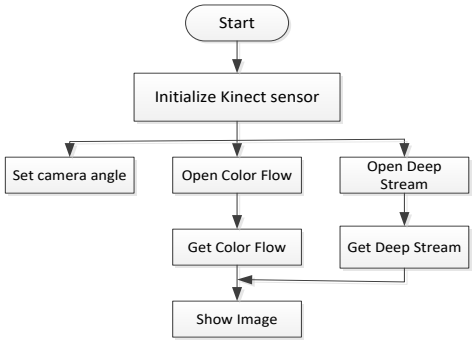

Figure. 1 Data initialization flow chart

  DOI: 10.25236/isrme.2019.026

The joint bilateral filter is a bilateral filter improvement algorithm that uses a range term and a color similar term. The RGB diagram and the depth diagram taken in the same scene are used as boot images and repair images, respectively. [4] According to the principle of joint bilateral filtering, the restoration formula of depth image is defined as

$$I(x, y) = \frac{1}{\omega_p} \sum_{i,j \in \Omega} \omega(i, j) D(i, j) \tag{1}$$

$I(x, y)$ is the processed image，$\mathrm{D}(i, j)$ is the initial image to be processed，Ω is the neighborhood of the currently processed pixels，$\omega(i, j)$ rights of the branch。ω_p is a standard quantity and can be represented by equation (2)

$$\omega_p \, \omega_p = \sum_{i,j \in \Omega} \omega(i, j) \tag{2}$$

The joint bilateral filter also pays attention to the similarity of pixels in the two domains of space and amplitude, so the weight $\omega(i, j)$ is expressed as:

$$\omega(i, j) = \omega_k(i, j) \times \omega_g(i, j) \tag{3}$$

The weight $\omega(i, j)$ is linearly related to the spatial distance of the pixel, and the nearest is the larger the weight $\omega_k(i, j)$ and its filter kernel function can be defined as follows:

$$\omega_k = (i, j) = \exp(-\frac{(i - x)^2 + (j - y)^2}{2\sigma_k^2}) \tag{4}$$

The weight $\omega(i, j)$ is linearly related to the pixel value. The pixel value at the edge of the image does not change much, and the weight value is relatively small. Its filter kernel function can be defined as follows

$$\omega_g(i, j) = \exp(-\frac{(G(i, j) - G(x, y))^2}{2\sigma_g^2}) \tag{5}$$

Among them,σ_k、σ_g are the standard deviations of $\omega_g(i, j)$ and $\omega_k(i, j)$ respectively.

3. System Model Building

3.1. Kinect and manipulator coordinate system establishment.

The Kinect and the manipulator coordinate system are represented by O_K-$X_K Y_K Z_K$, O_R − $X_R Y_R Z_R$ respectively. The position relationship between the two coordinate systems is shown in Figure 2.

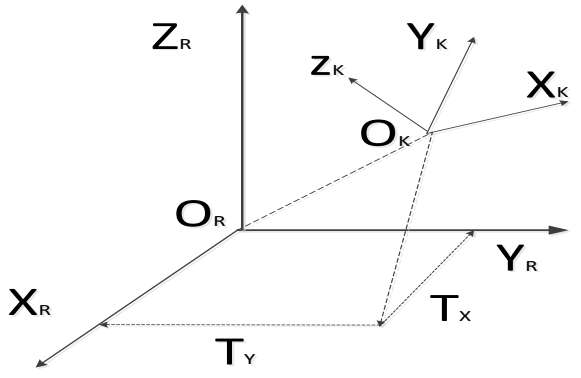

Figure. 2 Location diagram

3.2. Three-dimensional coordinate conversion principle.

In Figure 2, the origin of the two coordinate systems is obviously inconsistent. There are 3 flat shifts of $\Delta T_x, \Delta T_y, \Delta T_z$, There is a large Euler angle between the coordinate axes, but only through a certain rotation translation. Can make the two coordinate systems consistent. Three-dimensional coordinate transformation model is a widely used model in robot vision and other aspects.[5] It can map the coordinates of the target point under the original coordinate system to the target coordinate system to achieve coordinate conversion. The most widely used coordinate conversion model is the Bursha parameter model. The conversion model expression is

$$\begin{matrix} X_R \\ Y_R \\ Z_R \end{matrix} \quad \begin{matrix} X_K \\ = (1+\lambda) R_{(\varepsilon)} Y_K \\ Z_K \end{matrix} \quad \begin{matrix} \Delta T_x \\ \Delta T_y \\ \Delta T_z \end{matrix} \tag{6}$$

$[X_K Y_K Z_K]^{\mathrm{T}}$ is the Kinect coordinate system coordinate, $[X_R Y_R Z_R]^{\mathrm{T}}$ the coordinates for the manipulator; λ is a scale factor; $[\Delta T_x \Delta T_y \Delta T_z]^{\mathrm{T}}$ represents a flat shift in the coordinate system; R(ε)is a rotating matrix and R(ε)=$[R(\varepsilon_z) R(\varepsilon_y) R(\varepsilon_x)]$, $\varepsilon_z, \varepsilon_y, \varepsilon_x$ represents the Euler angle corresponding to the two coordinates respectively; defines d_x =($\Delta T_x \Delta T_y \Delta T_z \varepsilon_z \varepsilon_y \varepsilon_x (1+\lambda)$) as a coordinate conversion parameter vector.[6]

Solving the coordinate conversion model is to solve the 7 un known parameter in d_x,and usually three or more common points with two sets of coordinates are required. Since the common point cannot avoid errors, the coordinate correction is introduced and the error equation of equation (6) is converted to equation (7)

$$\frac{e}{3n \times 1} = \frac{A}{3n \times 7} \bullet \frac{d_x}{7 \times 1} + \frac{l}{3n \times 1} \tag{7}$$

In the formula: n represents the number of common points; a is the coordinate conversion coefficient matrix; L is a two-coordinate observation vector.

$$\frac{l}{3n \times 1} = \left[X_K - X_R, Y_K - Y_R, Z_K - Z_R\right]^T \tag{8}$$

When n>3, the coefficient matrix A often has errors, then introduces the overall least squares model

$$e - 1 = (A - E_A) \times d_x \tag{9}$$

e . represents an error vector; E_A is the error matrix of the coefficient matrix A. In order to solve the parameter vector d_x, the most important is to restore the E_A of the influence matrix A, but it is still difficult to solve it directly; Therefore, this paper adopts a linear over all least squares algorithm (LDLS) to perform linear processing of equation (9). It simplifies the calculation process and ensures the conversion accuracy. Here, provide n observers L_1, L_2, L_3, …, L_n used to estimate T independent variables.

$$\begin{aligned} L_1 &= L_1 + V_1 \\ L_2 &= L_2 + V_2 \\ &\vdots \\ L_n &= L_n + V_n \end{aligned} \tag{10}$$

Where V_i is the remaining vector observer. If n observers are linear models,

$$F_1(x_0 + \xi) L_1 + F_1(x_0 + \xi) L_2 + F_2(x_0 + \xi) V_2 + \cdots + F_n(x_0 + \xi) V_n + F_0(x_0 + \xi) = 0 \tag{11}$$

In the formula: x_0 is a variable approximation obtained by the nonlinear total least square

method; Indicates the approximate correction of the estimate, when $F_i(x_0+\xi)$ is a nonlinear equation, use the Taylor series to find the first derivative of x_0.

$$F_1(x_0)L_1 + F_1^{'}(x_0)L_1\xi + F_1(x_0+\xi)V_1 + \cdots + F_n(x_0)L_n + F_n^{'}(x_0)L_n\xi + F_n(x_0+\xi)V_n + F_0(x_0) + F_0^{'}(x_0)\xi = 0 \tag{12}$$

Formula (12) is another form of equation (9) that covers the residual of all observers. When all observers are independent, the objective function can be denoted as:

$$\Phi = V_1^T P_1 V_1 + \ldots + V_n^T P_n V_n - 2K^T(A\xi + BV + L) = \min \tag{13}$$

In the formula: P is the weight matrix of the observed value; K represents the Lagrange operator vector of m × 1. Finally, the necessary equations are obtained through the Euler-Lagrange necessary conditions

$$\frac{\partial\Phi}{\partial V_1} = 0, \frac{\partial\Phi}{\partial V_2} = 0, \cdots \frac{\partial\Phi}{\partial V_n}, \frac{\partial\Phi}{\partial\xi} = 0 \tag{14}$$

Integration of equations (12) and (13) with equation (14)

$$d_x = \left[A^T P_i (\sum_{i=1}^{n} F_i^{\,2}(x_0+\xi))^{-1} A\right]^{-1} \bullet \left[A^T P_i (\sum_{i=1}^{n} F_i^{\,x}(x_0+\xi))^{-1} L\right] \tag{15}$$

According to (15), all values of the parameter vector d_x can be solved, so the coordinate conversion model can be solved.

4. Location of grinding points

LTLS algorithm is used to solve coordinate transformation model. As shown in Figure 2, After two rotations, the angle of the two coordinate systems corresponding to the axis satisfies the linear relationship of small angles. The conversion matrix T is

$$T = B_x(-\frac{\pi}{2})R_z(\frac{\pi}{2}) \tag{16}$$

$$\left[X_K^{'} Y_K^{'} Z_K^{'}\right]^T = T\left[X_K Y_K Z_K\right] = \left[-Z_K - X_K Y_K\right] \tag{17}$$

By image processing of the information collected by Kinect, the initial coordinate value of the common point in the Kinect coordinate system is obtained.$[X_K Y_K Z_K]$T, Then calculate the common point coordinate value actually used for coordinate conversion$\left[X_K^{'} Y_K^{'} Z_K^{'}\right]$T from formula (16) and formula (17) .[7] Common point coordinates in the robot coordinate system$[X_R X_R X_R]$T can be obtained by robot position sensors and accurate measurements.

5. Experiments and Analysis

5.1. The specific process of the experiment is as follows.

The Kinect device is placed in a horizontal position to obtain a depth image of the target object and the manipulator, and uses an RGB color to represent different depths, de-noising the image, and repairing the three-dimensional coordinates of the target object and the manipulator. The distance between the manipulator and the target object is calculated through coordinate conversion and the data is transmitted to the manipulator. The manipulator uses a closed-loop control system. If the distance error between the manipulator and the grinding point exceeds 1cm, the LTLS algorithm is used again to adjust the position until the error is less than 1cm.

5.2. Analysis of experimental results.

Through experiments, it is proved that using the image generated by Kinect and the three-dimensional coordinate conversion, the manipulator can accurately perceive the unknown environment and can accurately move to the surface of the object for grinding operation.

6. Conclusion

A manipulator perception system based on Kinect depth sensor is designed, which can accurately convert the coordinates of the points to be ground in Kinect coordinate system to the manipulator coordinate system. In the manipulator system, the experimental results show that the method can achieve high precision perception location in a certain space.

References

[1] G.F. Li, L.L. Zhang, Y. Sun, J.Y. Kong. Multimedia Tools & Applications, 2018(1):1-18.

[2] D. Jiang, Z.J. Zheng, G.F. Li. Cluster Computing, 2018(3):1-11.

[3] G.F. Li, H. Tang, Y. Sun. Cluster Computing, 2017(3):1-11.

[4] Y. He, G.F. Li, Y.J. Liao. Cluster Computing, 2018(1):1-12.

[5] L. Lu, J.L. Zhang, Y.J. Zhu, H. Liu. Journal of Computer-Aided Design & Computer Graphics, 2015, 27(12): 2410-2418.

[6] D.J. Zhang, J.X. Li, Y. Zhang, Z. Zeng. Electric Machines and Control, 2014, 18(8): 87-93.

[7] B. Li, Y. Sun, G.F. Li. Cluster Computing, 2017(3):1-10.