Research on Object Location Detection based on ROS Machine Vision

Hong He, Siyang Wang, Shichao Ma

Tianjin University of Technology, Tianjin complex control research Laboratory,Tianjin 300384, China
E-mail: zysyj@qq.com

Abstract: Aiming at the problem that the positioning accuracy of machine vision is not high enough and low speed, this paper studies the object location detection based on ROS machine vision.Under the ROS platform, point cloud images are collected by Kinect, and multiple targets in complex scenes are identified and extracted by using MATLAB and OpenCV.In the early stage, the original point cloud image was processed by filtering and noise reduction and the improved target recognition method of single point cloud image was obtained through identifying part and edge detection of vector mutation in point cloud image. The experimental results show that the method can effectively extract the objects in the point cloud image, retain more details and feature points, calculate the geometric size, and facilitate the positioning and extraction in practical applications.

Key Words: Kinect, Point Cloud, Location, Object Detection , MATLAB

1 INTRODUCTION

The application of robots in various fields has been more and more extensive, but the problem of object orientation is still an urgent problem in this field[1]. Previously, Shen Jun and Zhang Hua of Southwest University of Science and Technology put forward a multidata fusion localization method based on ROS[2], and Sun Xiaokai of Zhejiang University studied an object localization and recognition method based on RGB-D information[3][4]. However, these localization methods require more data and a larger amount of calculation.

The emergence of Kinect simplifies this issue. Through the combination of depth camera Kinect device and ROS robot operating system, in the case of using a single point cloud image data, MATLAB and OpenCV filtering and denoising after operation, improved ICP characteristic vector detection algorithm is implemented for a single plane object detection and extraction, and high precision, achieved centimeter level, and that the volume of a solution to calculate the object and position, and other geometry information, effective use of point cloud data, have higher practical value.

2 PLATFORM CONSTRUCTION

In the vision-based ROS positioning problem, Kinect V2 can obtain RGB images and Depth images of the environment,simplify the acquisition of the depth of positioning. The Kinect V2 is equipped with a 1080p full HD wide-angle camera, which improves the depth fidelity three times and greatly improves the clarity of the identification of small objects and provides depth data stream information.[5]

By connecting to a PC running a ROS system and installing driver and publishing nodes in ROS,[10] the system resources can be effectively saved compared to the Windows development suite, and the original depth data and original point cloud data can be obtained more quickly, and the point cloud data can be extracted more easily.[6]

2.1 Kinect Imaging Principle

Kinect is a computational vision sensor that can capture both depth and color images. The positioning principle based on Kinect is based on the sequence of images captured. The positioning principle of Kinect is shown in Fig.1.

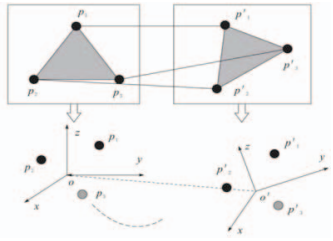


Fig1. Kinect Coordinate transformation

The triangle in Fig1 shows the image of the environment collected at different moments, and the feature extraction and matching are carried out. After the image matching, two matching feature point sets are obtained, which are three-dimensional coordinate points in two coordinate systems[7]. The point p_1 is in the $o-xyz$ coordinate system, the matching feature point $P_1^{'}$ is in the coordinate $o^{'}-xyz$ and the conversion relationship of the two points is (1):

This paper is supported by the key technologies R&D program of Tianjin(18YFZCSF00600).

$$\begin{bmatrix} x_1^{'} \\ y_1^{'} \\ z_1^{'} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = R_{3\times3} \begin{bmatrix} t_x \\ t_y \\ t_y \end{bmatrix} + t_{3\times1} \quad (1)$$

At least four pairs of collinear matching feature points are needed to obtain the transformation relationship between the corresponding feature points set of adjacent frames. The rotation matrix R and the translation matrix t are solved , namely the transformation matrix of pose P_j at the moment j relative to pose P_i at the moment I ,are obtained as (2)

$$T_i^j = \begin{bmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \quad (2)$$

The positional pose P_j at the moment j can be obtained from the positional pose P_i and transformation matrix T at the moment I as (3) :

$$P_j = P_i T_i^j = P_i \begin{bmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \quad (3)$$

The data at any time can be calculated by (4):

$$T_0^1 T_0^k = T_0^1 \times T_1^2 \cdots T_{k-1}^k \quad (4)$$

2.2 Calibration of the Kinect Camera

In order to achieve accurate positioning of the object, the Kinect camera needs to be calibrated before the experiment. For Kinect calibration, including camera calib- ration, the fusion of the depth image and the RGB image, the calibration toolbox in OpenCV can be used to correct the errors of the images. The use of Zhang Zhengyou checkerboard calibration method can simplify the complexity of the calibration problem[8][9]. In the calibration time, the camera model is pinhole model, and Zhang Zheng you model calibration method is defined as (5):

$$sm = A[R \quad t]M \quad (5)$$

It can be expressed as (6) :

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_{11} \\ r_{21} & r_{22} & r_{12} & t_{12} \\ r_{31} & r_{32} & r_{33} & t_{13} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (6)$$

In the above equation, (X_W, Y_W, Z_W) is the is one point coordinate in the camera coordinate system; (u, v) is the coordinates of (X_W, Y_W, Z_W) point projected on the image plane; A is the internal camera parameter matrix; $[R \quad t]$ is the external parameter matrix, R is the rotation matrix, and t is the translation matrix[10]. M is the homogeneous coordinates of one point in the world coordinate system; f_x, f_y are the focal length of the camera on the x and y axis respectively; c_x, c_y are the offset values of camera focus and imaging plane center respectively.

The inner parameter matrix is to describe the relationship between the camera coordinate system and the image coordinate system. The inner parameter can be used to calculate the point of the image coordinate system corresponding to the camera coordinate system, which is calculated as (7), (8):

$$X_C = \frac{(u - c_x)z_c}{f_x} \quad (7)$$

$$Y_C = \frac{(u - c_y)z_c}{f_y} \quad (8)$$

In the above formula, (X_c, Y_c) is the coordinate of one point in the RGB image in the camera coordinate system, and z_c is the depth value in the depth image corresponding to the point[11][12]. However, the color image and depth image acquired by Kinect V2 are not completely coincident, so it is necessary to match the color image and the depth image. Registration of Kinect color images and depth images is done through functions provided by OpenNI.[16]

3 PCOCESSING AND ALGORITHM OF POINT CLOUD DATA

3.1 Filter Selection

In the selection of the filter, the bilateral filter can be used to perform the edge denoising, and the bilateral filter is composed of two functions. One function is to determine the filter coefficients from the geometric spatial distance, and the other is to determine the filter coefficients from the pixel difference. Reducing the weights in the places where the difference is higher can obviously wrap the edges.

The bilateral filter has one more Gaussian variance $sigma - d$ than the Gaussian filter. The typical kernel function is Gaussian distribution function as (9):

$$\tilde{f}(x) = \eta^{-1} \int_\Omega w_{\sigma_s}(y) \Phi_{\sigma_r}(f(y) - f(x)) f(y) dy \quad (9)$$

$$\eta = \int_\Omega w_{\sigma_s}(y) \Phi_{\sigma_r}(f(y) - f(x)) dy \quad (10)$$

(10) is normalization, σ_s is the standard deviation of the spatial Gaussian function, σ_r is the standard deviation of the range Gaussian function, and Ω is the domain of the convolution. It can be seen that the value of $f(y) - f(x)$ changes very little in the flat region of the image, and the corresponding range weight is close to 1. At this time, the spatial weight plays a major role, which is equivalent to Gaussian blurring directly in this region. In the edge region $f(y) - f(x)$ have a large difference and

the range coefficient will decline, resulting in the decline of the whole kernel function distribution and keep the details of the edges.

In the bilateral filter, the value of the output pixel depends on the weighted combination of the values of the neighborhood pixels as (11):

$$g(i,j)=\frac{\sum_{k,l} f(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)} \tag{11}$$

The weight coefficient $w(i,j,k,l)$ depends on the core of the definition domain as (12):

$$d(i,j,k,l)=\exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2}\right) \tag{12}$$

The range core as (13):

$$r(i,j,k,l)=\exp\left(-\frac{\|f(i,j)-f(k,l)\|^2}{2\sigma_r^2}\right) \tag{13}$$

Multiplication result as (14):

$$w(i,j,k,l)=\exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2}-\frac{\|f(i,j)-f(k,l)\|^2}{2\sigma_r^2}\right) \tag{14}$$

It can be seen from the above formula that bilateral filtering takes into account the difference between the spatial domain and the range of values. The bilateral filtering algorithm of planar RGB image can be extend-ed to 3D plane to filter the point cloud. Firstly, [11]make the 3D point clouds meshed. For feature point clouds and non-feature point clouds, use different bilateral filtering factors to calculate the point cloud data ,and then denoising , which can prevent excessive smoothing caused by bilateral filtering that does not take into account the characteristics of the neighborhood points.

The bilateral filtering is defined as $P^{'}=P-\alpha n$, where P is the origin cloud point, $P^{'}$ is the denoised point, n is the direction of the normal vector, and α is the bilateral filtering factor, expressed as (15):

$$\alpha=\frac{\sum_{j=1}^{N} W_c\left(\|p-p_j\|\right)W_s\left(\|p-p_j,n\|\right)\langle p-p_j,n\rangle}{\sum_{j=1}^{N} W_c\left(\|p-p_j\|\right)W_s\left(\|p-p_j,n\|\right)} \tag{15}$$

The $W_c(x)=e^{-x^2/(2\sigma_c^2)}$ is defined smoothing weight function , and the $W_s(x)=e^{-x^2/(2\sigma_s^2)}$ is defined feature retains the weight function. Among them, σ_c , σ_s are Gaussian filter coefficients on the tangent plane, which respectively reflect the influence range of the tangential direction and the normal direction when calculating the bilateral filter function value of any point.[5]

The parameter σ_c is the influence factor of the distance from p to each neighborhood point to the point. The larger the σ_c is, the more the selected neighborhood points are. The parameter σ_s is the projection of the distance vector from the data point p to the adjacent point in the point method. The influence factor on the data point p ,the larger the σ_s is, the larger the distance at which the smoothing point p moves in the normal direction.

3.2 Improved ICP Algorithm

Point cloud matching is an automatic matching of two point clouds by using the dislocation characteristics between two point clouds through certain algorithms or statistical data rules. In the point cloud calibration processing, the initial matching is performed first and then the exact matching is performed**Error! Reference source not found.**. The accurate matching can effectively reduce the alignment error between two point cloud data. Iterative nearest point method is the most commonly used algorithm for accurate matching. The basic principle is to find the nearest point (p_i,q_i) in target cloud p and source cloud Q with matching according to certain constraints, and then calculate the optimal matching parameters R and t to minimize the error function. The error function is $E(R,t)$ as (16):

$$E(R,t)=\frac{1}{n}\sum_{i=1}^{n}\|q_i-(Rp_i+t)\|^2 \tag{16}$$

Where n is the number of nearest neighbor pairs, p_i is a point in the target point cloud P , q_i is the nearest point corresponding to p_i in the source point cloud Q , R is the rotation matrix, and t is the translation vector. Finally, it is judged whether the error converges, and when the error converges, the target point cloud P coordinate can be changed.

The existing ICP algorithms require the inclusion relationship between the two cloud data to be matched theoretically, but in fact the original data is difficult to meet this requirement, and each iteration of the ICP needs to search for the point, the calculation time is long[14].

The point cloud data model acquired by Kinect is composed of multi-point surfaces. By calculating the curvature value of the point in the point cloud, it is judged whether the point is a feature point, that is, a point located where the curvature or the normal vector is abruptly defined as a feature point. When searching for feature points, first need to set a surface average curvature threshold, and calculate the average curvature of each point[15]. If the average curvature is higher than the set threshold, the point can be used as a candidate feature point. Then calculate the feature vector curvature values of the candidate feature points wherein the feature vector curvature extreme points are actual feature points[18].

By comparing the experimental results with the traditional ICP algorithm, the improved ICP algorithm can effectively improve the accuracy of point cloud matching and shorten the time required for point cloud matching.

4 EXPERIMENTAL STEPS AND RESULTS PROCESSING AND ANALYSIS

Overall system implementation steps:

1. Collect the point cloud image using Kinect in the ROS environment of Linux system and take it as the original point cloud image.
2. Use the PCL point cloud library under Linux to perform image reading, denoise and filter the original point cloud data, and retain the RGB color data in the original point cloud image.
3. In MATLAB, the improved algorithm is used to extract the target point cloud image, including a series of important geometric information such as geometric contour.
4. Contrast the results of the unimproved algorithm and draw conclusions.

Perform feature calculation on the extracted object and iteratively calculate each point as (23)~(27):

$$x = \sum_0^N \left(x_{n_{\max}} - x_{n_{\min}} \right) \tag{23}$$

$$y = \sum_0^N \left(y_{n_{\max}} - y_{n_{\min}} \right) \tag{24}$$

$$z = \sum_0^N \left(z_{n_{\max}} - z_{n_{\min}} \right) \tag{25}$$

$$\theta = \arctan \frac{z_{\min}}{x} \tag{26}$$

$$l_{short} = \left(z_{\max} - z_{\min} \right) \sin\theta \tag{27}$$

The volume of the object calculated from the point cloud image is as shown in equation (28):

$$V_{cloud} = \left(x \sin\theta \right) \left(\left(z_{\max} - z_{\min} \right) \sin\theta \right) \left(y_l - y_r \right) \tag{28}$$

Where N is the number of coordinate points of each coordinate axis, $x_{\max}$, $x_{\min}$ is the maximum and minimum offset around a point, the y-axis is calculated in the same way, $z_{\min}$ is the shortest distance in the z-axis direction, and the ground deflection angle θ is solved, passing the ground bias The angle of the object in the z-axis direction is obtained from the angle.

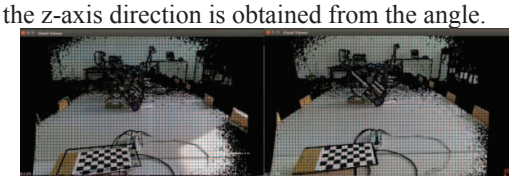


Fig2. Before and after Kinect calibration

The original point cloud data image acquired by Kinect in the ROS system is shown in Fig3.

Fig3. Original PointCloud Image

The original depth data image is shown in Fig4.

Fig4. Depth image

The algorithm proposed in this paper only uses the depth image and the original point cloud data. The original point cloud data contains RGB color information, and the depth image data can be used to improve the accuracy of the object depth information and preserve more contours information , using the improved algorithm in MATLAB software to extract objects in point cloud images as shown in Fig. 5. Fig6.

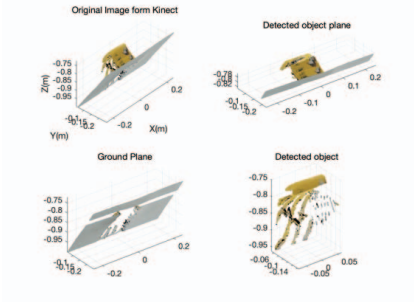


Fig5. Result from MATLAB (1)

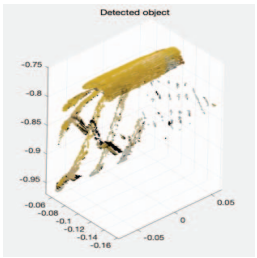


Fig6. Result from MATLAB (2)

Use point cloud computing data to obtain the geo-metric volume of the object and the actual measurement comparison chart as shown in Fig.7:

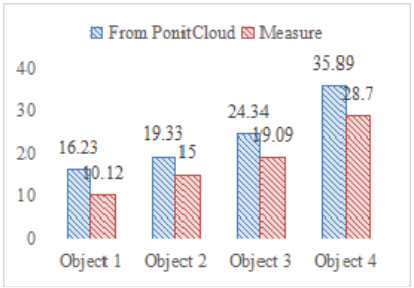


Fig7. volume compision chart

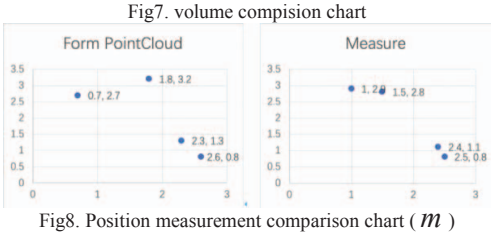


Fig8. Position measurement comparison chart (m)

It is calculated that the overall volume accuracy is between 62% and 79%. The smaller the object, the

larger the error between the actual measurement volume an-d the point cloud calculation volume, with the volume of the object increases, the accuracy gradually increase-es. In terms of positioning accuracy, the same situation is also presented. Although there are still errors, by merging other positioning algorithms, the subsequent reduction or elimination can be achieved, and the overall application can be achieved.

5 CONCLUSION

This paper proposes a method of positioning with a single point cloud image combined with RGB images, which effectively reduces the amount of data in the calculation of point cloud data, so that the positioning speed is improved and the accuracy is guaranteed. At the same time, the geometric information of the object is obtained. For the point cloud geometric boundary target of any deflection angle, the geometric target can be calculated and the error is within the acceptable range. Through the feasibility and accuracy of the method, the implementation speed is fast, the code quantity is small, and it has practical application value.

REFERENCES

[1] Li SR, Li Q, Li HY, Hou PH, Cao WG, Wang XD, Li H. Real-Time accurate 3D reconstruction based on Kinect V2. Ruan Jian Xue Bao/Journal of Software,2016,27(10):2519-2529 (in Chinese).

[2] Shen Jun, Zhang Hua, Xiao Yufeng, Chu Hongyu. Multi-data fusion location method based on ROS [A]. Light Engineering Technology, 2016, 1:83-92

[3] Li SR, Tao KL, Wang SY, Li HY, Cao WG, Li H. 3D reconstruction by Kinect sensor a brief review. Computer Aided Drafiting, Design and Manufacturing, 2014,24(1):1–11.

[4] Sun Xiaokai. Object location and recognition based on RGB-D information [D]. Zhejiang University, 2014.

[5] DING Meikun, XU Yulin, JIANG Caijun, RAN Peng. Kinect-based object grasping by robot arm hand system. J.Shanghai University (Natural Science Edition) , 2016, 22(4): 421-431.

[6] Dong Li Ya. Location detection of 3D objects based on Kinect [D]. Tianjin University of Science and Technology, 2015

[7] Zhang Yun, Gao Junchai, Xu Shumao. Kinect-based visual localization algorithm in complex environments [A]. Machinery and Electronics, 1001-2257 (2017) 11-0072-04

[8] Dora. Research on robot localization and recognition based on depth perception [D]. Nankai University, 2014.

[9] Zhou Zhen, Du Shanshan. Target localization and recognition based on Kinect depth images [B]. Electricity and automation, 1671-5276 (2016) 04-0173-04

[10]Zhang Songwei.ROS-based motion control and visual positioning of a four-axis manipulator[D].University of Chinese Academy of Sciences, 2018

[11]Berger K, Meister S, Nair R, Kondermann D. A state of the art report on Kinect sensor setups in computer vision. In: Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications. Berlin, Heidelberg: Springer-Verlag, 2013. 257–272. [doi: 10.1007/978-3-642-44964-2_12]

[12]Yang Jie. Research on robot localization algorithm integrating IMU and Kinect [D]. Wuhan University of Science and Technology, 2014

[13]Liu Dongqiu, Jing Fengxuan, Xie Xiaoyao.Improved ICP algorithm based on feature point extraction[A].Journal of Guizhou Normal University.2013, 12:106-110.

[14]Yang Hong, Qian Gui, Dai Xianzhong, et al. Creation of 3D indoor environment map of mobile robot based on Kinect sensor [J].Journal of Southeast University (Natural Science Edition), 2013, 43 (Supplement 1): 183-187.

[15]Chen Xiaoming, Jiang Letian, Lonicera japonica. Research on real-time 3D reconstruction and filtering algorithm based on Kinect depth information [J].Computer application research, 2013, 30 (4): 1216-1218.

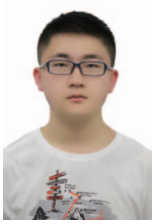
[16]PCL.Point Cloud Library. 2014.http://www.pointclouds.org/2014

[17]Liu Zhonghui.Research on image feature point descriptor algorithm based on KINECT sensor[D].Xi'an University of Electronic Science and Technology, 2014

[18]Rotation. Calibration and its application in augmented reality [D]. Hangzhou:Dong Chenmin, Chen Weihai, Yue Haosong et al. Automatic identification and localization of tomatoes based on Kinect vision system [J].China Agricultural Mechanochemistry Journal, 2014, 35 (4): 170-173. Zhejiang University, 2014.

Author's brief introduction

E-mail: :heho604300@126.com
Hong He : female, born in 1960, professor M.Sc. Tutor. Her main research direction is automation devices, object recognition and mobile robot navigation.



E-mail: :zysyj@qq.com
Phone number:(+86)15822761689
SiyangWang : (Corresponding author) Male. He is an undergraduate at Tianjin University of Technology now.His main research direction is computer vision, object location detection.

E-mail: 2522716234@qq.com
Shichao Ma: female. She is a M.Sc. candidate in Tianjin University of Technology now. Her main research direction is grasping with mechanical arm and the depth of the visual.