Pose Estimation of Rigid Object in Point Cloud

LIU Zongming
School of Electrical Engineering and Automation
Harbin Institute of Technology, Harbin, China
Shanghai Institute of Spaceflight Control Technology, Shanghai, China
Shanghai Key Laboratory of Space Intelligent Control Technology, Shanghai,, China

LI Jianxun
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China

LIU Guodong
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China

YE Dong
School of Electrical Engineering and Automation
Harbin Institute of Technology
Harbin, China

Abstract—**Using (160×120) 3D point cloud dataset, we here propose an approach to estimate 6D pose of rigid objects at runtime frequency of 33fps. This approach is useful for tracking object's pose during object uniform rotation. Firstly, removing outliers using a StatiscalOutlierRemoval filter. Then, Euclidean cluster extraction is made for detecting the object. Thirdly, ICP algorithm is in process between two adjacent images, computing the transformation matrix. Finally, we can learn the rotation angle of the object. Experiments show that the method has good performance.**

Keywords- Point Cloud, Euclidean cluster, ICP, 6D pose

I. INTRODUCTION

6D Pose estimation is an important and challenging task in the field of robot vision, which is widely used in military guidance, visual navigation, robot, intelligent transportation, public safety and son on. In the fields of driving development in this area, robotics, in particular, has a powerful need for computationally efficient approaches, as the accurate information about the structure of the scene and the objects are required to perform operational tasks.

One of the biggest limit of a normal camera is the difficulty of extracting depth information from the recorded data. However, in the last year, the depth camera has become a popular and accessible tool for acquiring information about the environment. Depth data can be got from stereovision camera, LIDAR device and so on. Using the depth information to estimate the object pose is available and more precisely. The research on the pose estimation issue involves a wide range of algorithms and approaches. The mainly vary according to the features used to describe the object, the mathematical approach to solve the problem and the format of the data given as input to the algorithm representing the model of the object to find. The localization of an object in a 3D scene is mainly addressed by using features descriptors which univocally identify an object or a part of the object. In particular with features we can refer both to the result of a neighborhood operation applied to the image (such as FPH and FPFH) [1] or to specific structures of the image (edges, corners, blobs, ridges, shape). However, sometimes the data acquired by 3D camera is pretty rough and the number of points on the object is small. Besides that, the resolution of point cloud data is low that the key point is not consistent. So, in this paper, using (160×120) resolution point cloud dataset, we present a method that can perform well for pose estimation. So, in this paper, we present a method based on iterative closest point algorithm. After obtaining the 3D point cloud of scene, we remove the outlier points using StaticalOutlierRemoval filter, which make more accurate registration of the object. Then, we detect the object from the background using Euclidean cluster algorithm. When object caught, the registration between two adjacent images is done, so that we can get the six degrees information of the target.

The paper is organized as follows. Section 2 introduces the principle of stereo vision and point cloud. Section 3 depicts the approach we proposed, from 3d segmentation of point cloud to ICP algorithm [2]. In section 4 experiments and results analysis are presented, and we conclude in section 5.

II. STEREO VISION

A point cloud [3] is a set of data points in some coordinate system, each point contains a three-dimensional coordinates, and some may contain color information or intensity information. With the advent of the new era, low-cost 3D sensing hardware such as Kinect, and the continuous efforts of advanced point cloud processing, 3D perception is getting more and more attention in stereo vision pose estimation.

At present, the method based on the principle of computer vision has become main stream technology in the field of 3D information retrieval. According to whether or not take the initiative light source lighting, it can be broadly divided into two categories: passive vision and active vision. Passive vision mainly includes monocular vision and stereo parallax method. Stereo parallax method is used to simulate the mechanism of human vision. Here, we introduce the principle of binocular stereo vision.

Binocular stereo vision theory [4] is based on the study of the human visual system. However it is stronger than the human eye in the aspect of depth information calculating.

978-1-5090-3710-0/16/$31.00 ©2016 IEEE

The realization of binocular stereo vision is based on the principle of parallax, and its model shows in Figure 1, which is a mathematical model of no distortion, alignment, and has been measured. The image planes of two cameras are precisely located on the same plane, the optical axis is strictly parallel, the distance is certain for T, and the focal length is the same $f_l = f_r = f$. In addition, the left main point c_x^{left} and the right main point c_x^{right} have been calibrated, That is the main point in the left and right images with the same pixel coordinates. In the model, the two images are aligned, and the line alignment means the two images are on the same plane, and each line is strictly aligned. Assuming that physical world point is $P(x, y, z)$, and the corresponding transverse coordinates are (x^l, y^l) and (x^r, y^r).

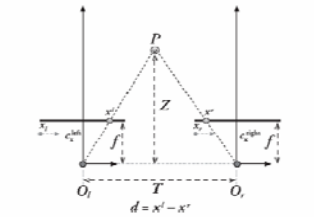


Figure 1. Binocular stereo vision

Here, we make the left camera coordinate system as the world reference coordinate system.

$$d_{x_l} = x^l - c_x^{left} \tag{1}$$

$$d_{x_r} = c_x^{right} - x^r \tag{2}$$

The depth Z value can be deduced by using similar triangles as:

$$\frac{fT}{x^l - x^r - (c_x^{left} - c_x^{right})} \tag{3}$$

We can calculate the x-axis coordinate and the y-axis coordinate similarly:

$$\begin{cases} X = \dfrac{x^l T}{x^l - x^r - (c_x^{left} - c_x^{right})} \\ Y = \dfrac{y^l T}{x^l - x^r - (c_x^{left} - c_x^{right})} \\ Z = \dfrac{fT}{x^l - x^r - (c_x^{left} - c_x^{right})} \end{cases} \tag{4}$$

Calculating the three-dimensional coordinates of each point, 3D point cloud is got.

Here, we use a 3D camera Fotonic E70. E70 is a 3D camera based on time of flight with high reliable performance and rugged physical design. It provides very low motion artifacts and a high frame rate, which can effectively track moving targets. In addition, Fotonic E70 can provide the best performance in outdoor lighting with high power illumination. The specific parameters are as follows:

TABLE I. FOTONIC PARAMETERS

Type of sensor	CCD
Maximum frame rate	58 fps
Total cpature time	7ms
Pixel array size	160(h) × 120(w)
Number of dead pixels on sensor	<= 20
Field of view (h) ×(v)	70° × 53°
CPU	1.5GHz Dual-core ARM Cortex-A9
Memory	512 MB 400MHz LPDDR2
OS	Linux
Drivers	Linux Windows XP/7/8
PC API	FZ-API for C, C++, PCL
Camera Internal API	FZ-API
Cross Compiler and Debugger	GCC and GDB for ARM Cortes
Distance data resolution	16bit/pixel
Signal amplitude resolution	10bit/pixel
Data interface	Gigabit Ethernet

The Fotonic E70 device:

Figure 2. Image of Fotonic E70

III. PROPOSED METHOD

Because the resolution of 3D camera is low, and the object surface texture is not abundant, the key point and feature descriptor is not consistent, which can easily lead to point cloud mismatch. Therefore, this paper propose a object pose estimation algorithm bases on ICP registration on adjacent frames.

Firstly, Filtering the scene point cloud, and remove noise. Secondly, extracting the object information based on Euclidean cluster method; and then, two successive frames registration is in process, which estimate the 6 degree of freedom information of object. Flow chart is shown below:

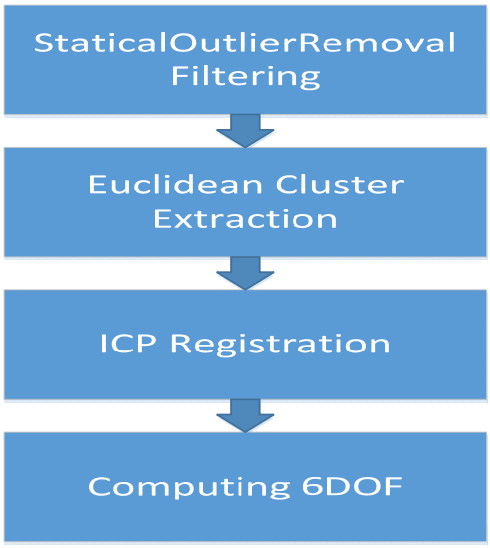


Figure 3. Flow Chart

A. 3D Segmentation of Point Cloud

There are two methods for point cloud segmentation: cluster segmentation and segmentation based on sampling consistency. Here, we make use of cluster segmentation for detecting object.

1) StaticalOutlierRemoval Filter[5]

Laser scanning usually produces an uneven density of point cloud data set. In addition, the error in the measurement can produce a sparse outlier, which makes the effect worse. The computation of local point cloud features (such as the normal vector or curvature change rate of the sampling points) is very complex, which can lead to erroneous values and in turn may lead to the failure of post processing of point cloud registration. The following methods can solve some of these problems: a statistical analysis of the neighborhood of each point, and pruning the points that do not conform to a certain standard. The StaticalOutlierRemoval method is based on the computation of the distance distribution to the point in the input data. For each point, we compute the average distance to which it is at its closest point. The result obtained is on the assumption of a Gauss distribution, its shape is determined by the deviation of the mean value and the standard. The average distance out of the standard range can be defined as outliers and removed from the data set.

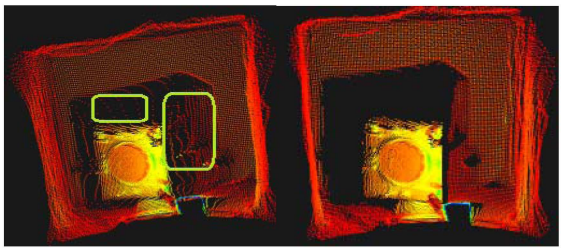


Figure 4. StaticalOutlierRemoval filter

In Figure 4, two pictures show the effect of sparse outlier removal; the left image if the original dataset and the right image is the resultant data set.

2) Euclidean Cluster Extraction

Point cloud segmentation is based on the space, geometry and texture characteristics of the point cloud, making the point cloud in the same division have similar characteristics. Effective segmentation of point cloud is the front of feature extraction for object. Commonly used point cloud segmentation algorithm are random sample consensus algorithm and cluster algorithm. Random sample consensus segmentation can only be focused on a specific point cloud data segmentation of a model, which does not apply to the scene with a number of point cloud clustering. Euclidean clustering segmentation algorithm is relatively easy to understand, that is, the distance in a certain threshold are considered as a class.

Here, we give the steps for Euclidean cluster extraction [6]:

- Create a kd-tree representation for the input point cloud dataset P.
- Set an empty clustering set C, as well as a queue Q to store the set of points to be tested.
- For any point $p_i \in P$, the following steps are performed:
  - Add p_i to the current queue Q.
  - For each point $p_i \in Q$ do:
    - With p_i as the center, Search for the set P_k^i in a sphere with radius $r < d_{th}$.
    - For each field $p_i^k \in P_i^k$, check if the point has already been processed, if not join the queue Q.
  - If all the points in Q are processed, then all points are put into the cluster C, and reset Q to an empty queue.
- When all points $p_i \in P$ have been processed, and a variety of clusters in C, the algorithm ends.

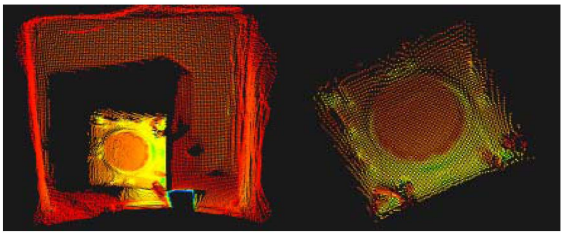


Figure 5. Euclidean cluster extraction

B. *Iterative Closest Point*

Iterative Closest Point (ICP) is the most used algorithm to align two point clouds. The algorithm requires as input two point clouds and give as output the rotation and translation required to overlap the two datasets in order to minimize the distance between them. Starting from rough estimation of the pose, it iterates following these major steps for each point:

- Find the closest point.
- Find the best transform for this correspondence.
- Transform the dataset

Given two point clouds M and P, for each point of P, the closest point in M is found according to the Euclidean distance, here, we can make a kd-tree to find the closest point in M. The mathematical formula which express these steps is the following one:

$$\frac{1}{M}\sum_{v\in M}\| v - match(v)\|^2 \tag{5}$$

Then, the transformations R and T which minimize

$$\sum_{v\in M}\| Rm_v - T - p_v\| \tag{6}$$

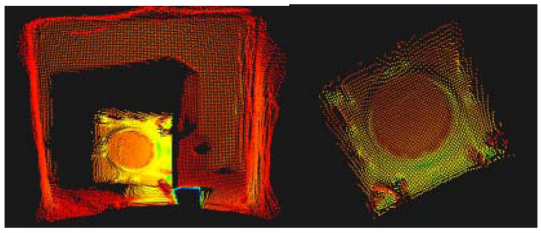
Are computed, where R is the 3D rotational matrix, T the 3D translation vector, $m_v \in M$ and $p_v \in P$. The ICP algorithm is guaranteed to converge to a local minimum, but if the initial guess is accurate enough then it converges to the global minimum. The error decreases monotonically.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

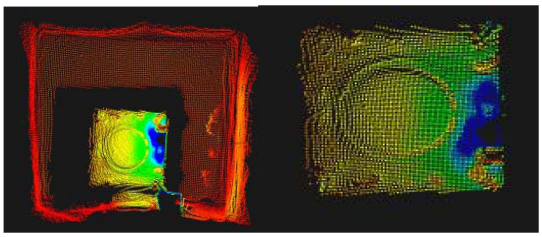
Our experiment platform is Visual Studio 2015 and PCL 1.8.0. PCL [7]. The 3D camera is Fotonic E48, which can acquire (160×120) resolution point cloud dataset. Here, the experiment is divided into four groups, which is based on the distance and the rotation speed of the object. The position of 3D camera is respectively 3 meters and 4 meters, and the rotation speed of object is respectively $2^\circ / s$ and $5^\circ / s$.

A. *3D Segmentation of Point Cloud*

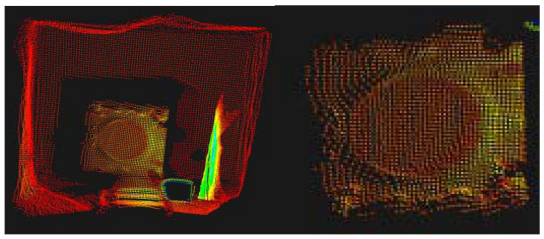
After StaticalOutlierRemoval filter and Euclidean Cluster Extraction, the object is detected from the scene point cloud dataset, as shown in Fig 4.



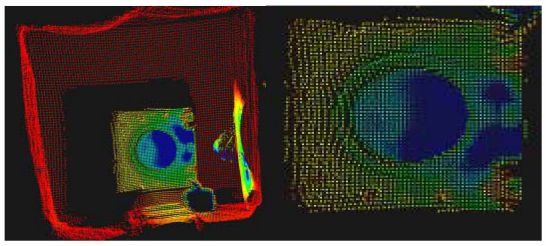
(a) 3 meters and $2^\circ / s$



(b) 3 meters and $5^\circ / s$



(c) 4 meters and $2^\circ / s$

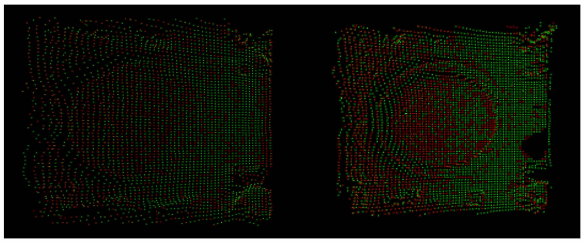


(d) 4 meters and $5^\circ / s$

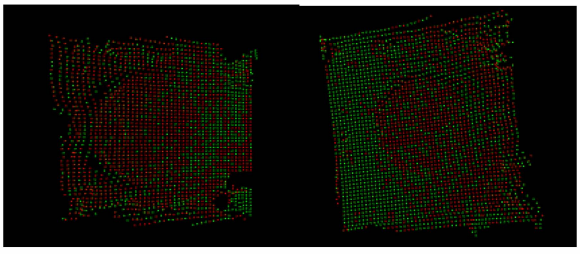
Figure 6. 3D object detection

B. *ICP Registration*

3D point clouds registration using ICP algorithm are shown in Figure 7.



(a) 3 meters and $2^\circ / s$ (b) 3 meters and $5^\circ / s$



(c) 4 meters and $2^\circ / s$ (d) 4 meters and $5^\circ / s$

Figure 7. 3D point cloud registration

In Figure 7, the red point cloud represents the pre frame, and the green point cloud represents the next frame. For (a), the transformation matrix:

$$H = \begin{matrix} 1 & 0 & -0.001 & 3.397 \\ 0 & 1 & 0 & -0.867 \\ 0.001 & 0 & 1 & -0.434 \\ 0 & 0 & 0 & 1 \end{matrix} \quad (7)$$

Then, the translation vector:

$$T = 3.397 \quad -0.867 \quad -0.434 \quad (8)$$

The rotation angles of X, Y, Z axis are:

$$(roll, pitch, yaw) = -0.0003 \quad -0.0011 \quad 0.0002 \quad (9)$$

For (b), the transformation matrix:

$$H = \begin{matrix} 1 & 0 & -0.003 & 9.068 \\ 0 & 1 & 0 & 1.091 \\ 0.003 & 0 & 1 & 0.051 \\ 0 & 0 & 0 & 1 \end{matrix} \quad (10)$$

The translation vector:

$$T = 9.068 \quad 1.091 \quad 0.051 \quad (11)$$

The rotation angles of X, Y, Z axis are:

$$(roll, pitch, yaw) = 0.0005 \quad -0.00308 \quad 0.0004 \quad (12)$$

For (c), the transformation matrix:

$$H = \begin{matrix} 1 & 0 & -0.001 & 4.039 \\ 0 & 1 & -0.001 & -0.403 \\ 0.001 & 0.001 & 1 & 0.460 \\ 0 & 0 & 0 & 1 \end{matrix} \quad (13)$$

The translation vector:

$$T = 4.039 \quad -0.403 \quad 0.460 \quad (14)$$

The rotation angles of X, Y, Z axis are:

$$(roll, pitch, yaw) = -0.0013 \quad -0.0012 \quad 0.0008 \quad (15)$$

For (d), the transformation matrix:

$$H = \begin{matrix} 1 & 0 & -0.003 & 11.146 \\ 0 & 1 & 0.001 & -2.108 \\ 0.003 & -0.001 & 1 & 2.716 \\ 0 & 0 & 0 & 1 \end{matrix} \quad (16)$$

The translation vector:

$$T = 11.146 \quad -2.108 \quad 2.716 \quad (17)$$

The rotation angles of X, Y, Z axis are:

$$(roll, pitch, yaw) = -0.00053 \quad -0.00288 \quad -0.00023 \quad (18)$$

The 3D camera frame rate is 33ftps, so we can learn the object rotation angle between two adjacent images. In (a), the angle of pitch is 1.91°; In (b), the angle of pitch is 5.2°; In (c), the angle of pitch is 2°; In (d), the angle of pitch is 5°.

Next, the fixed rotation angle pose estimation is done. At the initial state of the object, we capture the point cloud information; then object rotate for 2 degrees and capturing the point cloud data. ICP registration on the two point cloud:

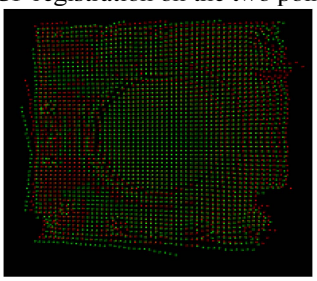


Figure 8 Registration for 2 degree

The transformation matrix is:

$$H = \begin{matrix} 1 & 0 & -0.035 & 126.59 \\ 0.001 & 1 & -0.0026 & 9.17 \\ 0.035 & 0.0026 & 1 & 23.49 \\ 0 & 0 & 0 & 1 \end{matrix} \quad (19)$$

The translation vector is:

$$T = 126.59 \quad 9.17 \quad 23.49 \quad (20)$$

The rotation angles of X, Y, Z axis are:

$$(roll, pitch, yaw) = 0.0026 \quad -0.0345 \quad -0.000002 \quad (21)$$

Therefore, the object's rotation angle around y-axis is 1.98 degree, the x-axis is 0.14 degree, and the z-axis is 0.0001 degree. As can be seen, the object pose estimation is close to its true value.

C. Statical Analysis

For each scenario the algorithm has been executed and the rotation angle around the y axis has been computed. We will show the pitch angle in continuous 50 frames and compare it with the true value.

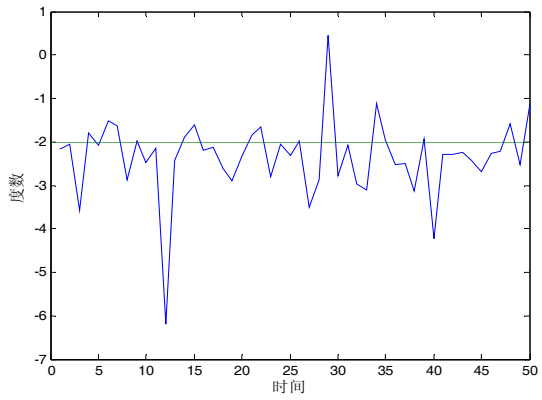
From all figures, we can see that the mean value of the rotation angle around the Y axis is equal to the true value. The algorithm proves perform well in low resolution point cloud dataset.

V. CONCLUSION

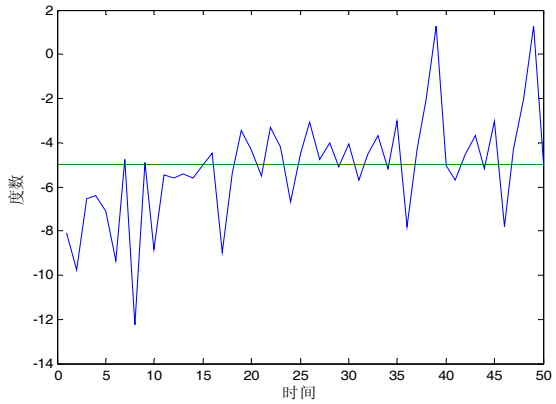
Stereo vision can facilitate depth perception such that we can obtain more object information. So, in this paper, we represent a method based on ICP algorithm to estimate the rigid object pose. Removing the outliers of the point cloud by

StaticalOutlierRemoval filter and then detect the object using Euclidean cluster extraction algorithm. After those, ICP algorithm is in process between two adjacent images. And we can get the 6D pose of the object.

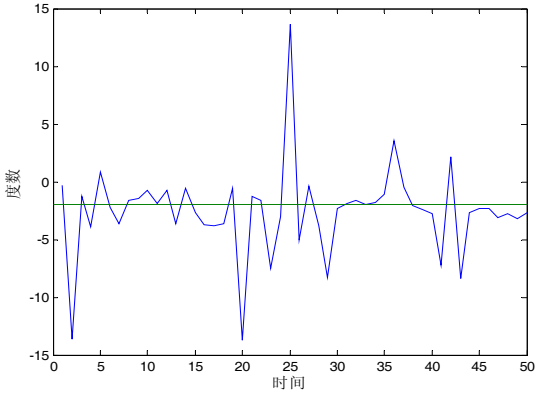
In conclusion, the proposed approach gives interesting results for point cloud whose resolution is low. In these cases it can represent an efficient alternative to other method, since it does not require key point. Besides that, it make use of depth information that it perform better than the 2D pose estimation.



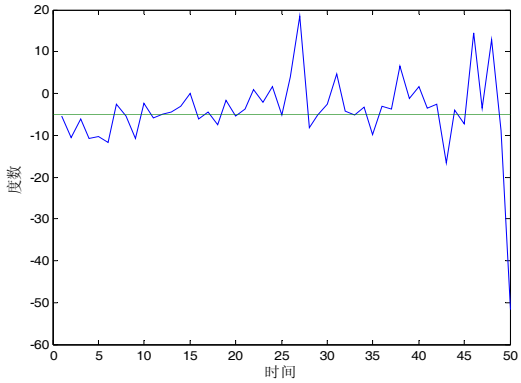
(a) 3 meters and $2^{\circ}/s$



(b) 3 meters and $5^{\circ}/s$



(c) 4 meters and $2^{\circ}/s$



(d) 4 meters and $5^{\circ}/s$

Figure 9. Rotation angle of the Y axis

ACKNOWLEDGMENT

This work was jointly supported by National Natural Science Foundation (61175008); Shanghai Academy of Spaceflight Technology - Shanghai Jiao Tong University Aerospace Advanced Technology Joint Research Center Fund (USCAST2015-8); Aerospace Science and Technology Innovation Fund and Aeronautical Science Foundation of China (20140157001); 2015 Industry-university-research cooperation project of AVIC; Shanghai Rising-Star Program (16QB1401000).

REFERENCES

[1] Rusu R B, Cousins S. 3d is here: Point cloud library (pcl) [C] Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011: 1-4.

[2] Estépar R S J, Brun A, Westin C F. Robust generalized total least squares iterative closest point registration[C] International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer Berlin Heidelberg, 2004: 234-241.

[3] Tamimi H, Andreasson H, Treptow A, et al. Localization of mobile robots with omnidirectional vision using particle filter and iterative sift[J]. Robotics and Autonomous Systems, 2006, 54(9): 758-765.

[4] Blake R, Wilson H. Binocular vision [J]. Vision research, 2011, 51(7): 754-770.

[5] Skinner B, Vidal-Calleja T, Miro J V, et al. 3D point cloud upsampling for accurate reconstruction of dense 2.5 D thickness maps[C]//Australas. Conf. Robot. Autom. (ACRA). 2014.

[6] Phan A, Ferrie F P. Towards 3D human posture estimation using multiple kinects despite self-contacts[C]//Machine Vision Applications (MVA), 2015 14th IAPR International Conference on. IEEE, 2015: 567-571.

[7] Rusu R B, Cousins S. 3d is here: Point cloud library (pcl) [C] Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011: 1-4.

[8] Guo F, He Y, Guan L. An improved ICP registration algorithm with a weight-bootstrap scheme[C] Multimedia Signal Processing (MMSP), 2015 IEEE 17th International Workshop on. IEEE, 2015: 1-6.

[9] Lv Dan, Sun Jianfeng, Li Qi, Wang Qi. 3D pose estimation of target based on ladar range image [J]. Infrared and Laser Engineering, 2015, 44(4): 1115-1120