Contents lists available at YXpublications

International Journal of Applied Mathematics in Control Engineering

Journal homepage: http://www.ijamce.com

Design and Implementation of Monocular Vision SLAM for Mobile Robot Based on ROS

Rui Peng[a], Lei Cheng[a,b,*], Yating Dai[a], Xitong Zhao[a] , Huaiyu Wu[a], Yang Chen[a]

[a] *School of information science and engineering, Wuhan university of science and technology, Wuhan 430081, China*

[b] *School of automation, Hangzhou dianzi university, Hangzhou 310000, China*

ARTICLE INFO

Article history:
Received 24 June 2017
Accepted 11 September 2017
Available online 28 October 2017

Keywords:
ROS;
mobile robot;
monocular vision;
SLAM

ABSTRACT

Aiming at mobile robot localization and mapping Room problems, this paper proposes a method based on ROS (Robot Operating System) monocular vision mobile robot SLAM (Simultaneous Localization And Mapping) method. The method is divided into five parts: monocular camera captures image information; visual odometry does estimation of camera motion and local map building; map of back-end optimization; using loop detection to eliminate the accumulated error; building maps based on existing information. Through the two ways of off-line data set test and real-time test, we get satisfactory results, and prove the feasibility and rationality of the method.

1. Introduction

The positioning and mapping of the mobile robot [1] means that the robot should understand its own state and comprehend the external environment, and the main problems are focused on the positioning. At present, mobile robot localization can be divided into two categories: one is sensors carried on the robot's body, such as robot's encoders, cameras, laser sensors and so on; the other is that sensors are installed in the environment, such as GPS, guide rail, two-dimensional code sign, and so on. The sensor equipment installed in the environment usually can directly measure the location information of the robot and solve the problem of positioning simply and effectively. However, as they have to be set in the environment, to a certain extent, the scope of the application of the robot is limited. For example, there are no GPS signals in some places, and some places are unable to lay a guide. So this kind of sensor constrains the external environment. Only when these constraints are met, their location schemes can work, so these sensors are simple and reliable, they cannot provide a universal and common solution. Relatively speaking, sensors carried on the robot's body can calculate their location indirectly by reading some observation technology in the external environment. It does not make any requirements for the environment, so that the location schemes can be applied to the unknown environment. Therefore, in this paper, a mobile robot monocular vision SLAM [2] method based on ROS is proposed to locate and map the experimental objects under unknown environment, and verify the correctness and feasibility of the method through theoretical analysis and experiment.

2. Introduction of ROS

The experimental platform of this paper is ROS. ROS is an open source robot operation system released by Willow Garage in 2007. It provides many excellent tools and libraries for software developers to develop robot applications. At the same time, there are also excellent developers who continue to contribute code to it. In essence, ROS is not a real operating system, but more like a software package based on an operating system. It provides a number of algorithms that may be encountered in real robots: navigation, communications, path planning, and so on. It supports the widely used object oriented programming language C++, as well as the scripting language Python.

ROS provides some standard operating system services, such as hardware abstraction, underlying device control, common functional implementations, inter process messages, and packet management. ROS is based on a graph structure, so that different nodes processes can accept, publish and aggregate various kinds of information, such as sensing, control, status, planning and so on. At present, ROS is mainly supported by Ubuntu.

ROS can be divided into two layers: the lower level is the operation system level described above; and the high-level is the various software packages that the majority of users contribute to

* Corresponding author.
E-mail addresses: chenglei@wust.edu.cn (Lei Chen)

achieve different functions, such as location mapping, action planning, perception, simulation etc.

ROS (low level) uses the BSD license, all of them are open source, and can be used for research and commercial use free of charge. The packages provided by a high level user can use a number of different licenses.

3. Monocular camera

3.1 Introduction of monocular camera

The camera used in SLAM is not the same thing as the single mirror camera we normally see. It is more simple, not to carry expensive lenses, but to shoot the surrounding environment at a certain rate and form a continuous video stream. According to the different mode of work, the camera can be divided into three categories: single camera (Monocular), binocular camera (Stereo) and deep camera (RGB-D)[3].

A camera that works with a single camera is called monocular camera. The structure of the sensor is very simple, and the cost is very low, but there are some problems. The picture is essentially a projection on the camera's imaging plane when the scene is photographed. It reflects the three dimensional world in a two-dimensional form. In this process, a dimension of the scene is lost, that is, the lack of depth information. In a monocular camera, we can't use a single picture to calculate the distance between the objects in the scene and us. That is, in single image, we cannot determine the true size of an object, it can be a great but far away objects and may also be a very close but very small objects, because near the small, they become the same size in the image. Therefore, monocular camera in SLAM[4]-[5] will lead to a scale factor (Scale) between our estimated trajectories and maps and the real trajectories and maps, that is, the scale uncertainty of monocular cameras.

3.2 Monocular camera model

The geometric model of a camera is the process of mapping a point (unit meter) in a three-dimensional space to a two-dimensional image plane (unit pixel). The monocular camera model[6] is the same as our common pinhole camera model, as shown in Figure 1.

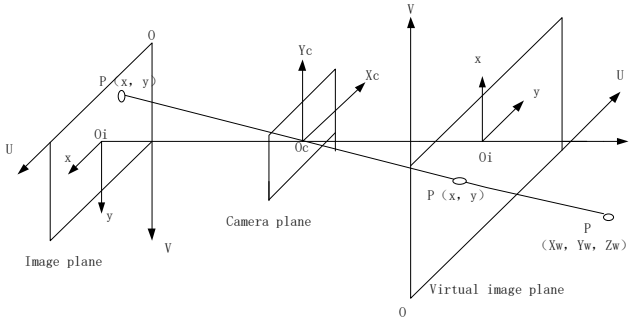

Figure 1 coordinate system

Set up camera coordinate system $O_c - x_c - y_c - z_c$, Physical coordinate system of image $O_i - x - y - z$, Pixel coordinate system $O - u - v$ And world coordinate system $O_w - x_w - y_w - z_w$.

1) Conversion of camera coordinate system to image physical coordinate system

Set the point of three-dimensional space $P_w(X_w, Y_w, Z_w)$ coordinate in the camera coordinate system $P[X_c, Y_c, Z_c]^T$ its projection falls on the physical imaging plane, point P Coordinates in the physical coordinates of the image $P_i[X, Y, Z]^T$, the distance between the plane of physical imaging and the hole is focal length f (unit meter), according to the principle of triangle similarity, the imaging plane is symmetrical to the front of the camera (the mathematical method for dealing with the real world and camera projection) to simplify the model:

$$\begin{cases} X = f\dfrac{X_c}{Z_c} \\ Y = f\dfrac{Y_c}{Z_c} \end{cases} \tag{1}$$

2)Conversion of image physical coordinate system to pixel coordinate system

In the image pixel coordinate system（the original point O is located at the upper left corner of the image, the u axial right is parallel to the axis x, the v axial right is parallel to the axis y）, the transformation relation of the point P_i to the point $P_{uv}(u_0, v_0)$:

$$\begin{cases} u_0 = \dfrac{X}{dx} + c_x \\ v_0 = \dfrac{Y}{dy} + c_y \end{cases} \tag{2}$$

In formula (2), dx, dy are the physical size of each pixel on x axis and y axis (per pixel per unit of meter) respectively; c_x, c_y are the coordinates of the u axis and v axis in the pixel coordinate system of the image physical coordinates system O_i (units are pixels) respectively.

3)Conversion of camera coordinate system to image pixel coordinate system

The transformation relationship between P points in camera coordinates and pixel coordinates $P_{uv}(u_0, v_0)$ in camera coordinates can be obtained from (1) to (2):

$$\begin{cases} u_0 = \dfrac{f}{dx}\dfrac{X_c}{Z_c} + c_x \\ v_0 = \dfrac{f}{dy}\dfrac{Y_c}{Z_c} + c_y \end{cases} \tag{3}$$

In the formula (3), $\frac{f}{dx}, \frac{f}{dy}$ are merged into $\frac{f}{dx}, \frac{f}{dy}$,(units of pixels) respectively, after finishing:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (4)$$

In the formula (4) , P is the homogeneous coordinates, the matrix K is the internal non parameter of the camera.

4)The transformation of the world coordinate system to the image pixel coordinate system

The coordinates of the point P in the world coordinate system are (X_w, Y_w, Z_w), the transformation matrix of the world coordinate system to the camera coordinate system is T, the transformation relation between the world coordinate system and the pixel coordinate system is as follows:

$$Z \begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} = K \left(R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + t \right) \quad (5)$$

The rotation matrix R and the translation vector t in the form (5) indicate that the position of the camera is called the external parameter of the camera. The external participant changes with the motion of the camera, and it is also an estimated target in the SLAM. This parameter represents the trajectory of the camera.

5)Normalization

Finally, the coordinates P are normalized and the projection point P_c on the camera normalization plane are obtained:

$$P_c = \begin{bmatrix} X_c / Z_c \\ Y_c / Z_c \\ 1 \end{bmatrix} \quad (6)$$

It is called the normalized coordinate, which is located on the plane Z = 1 in front of the camera, which is called the normalized plane.

4. Monocular vision SLAM

4.1 Visual SLAM

The whole visual SLAM includes the following steps:

1)Sensor information reading:

In visual SLAM, it mainly reads and preprocesses the image information of the camera.

2)Visual Odometry:

The visual odometer is to estimate the motion of a camera between adjacent images, as well as the appearance of a local map.

3)Back end optimization:

The backend accepts the camera pose measured at different time, and the information of loopback detection, optimizes them, and gets the globally consistent track and map.

4)Loop detection:

Loop detection determines whether the robot has reached the previous position. If the loop is detected, it will provide information to the back end for processing.

5)Composition:

A map corresponding to the task requirements is established according to the estimated trajectory.

The following figure2 is the framework for visual SLAM:

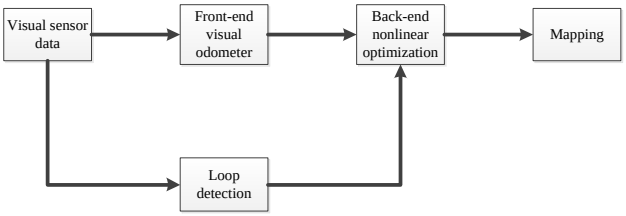

Figure 2 framework for visual SLAM

4.2 The mathematical description of visual SLAM

To simplify the problem, we divide the visual SLAM[7] into two parts: the motion part and the map section. Because cameras usually collect data at certain times, in motion part, we can turn a period of motion into something $t = 1, \cdots, K$ in discrete time. At these moments, x is used to represent the position of the robot. The positions $x_1, \cdots, x_K$ at all times are recorded as the trajectories of the robot. In the map part, we assume that the map is made up of many road signs. At any time, the camera will measure part of the road punctuation and get their observation data. Then the number of path punctuation is N, which is expressed by $y_1, \cdots, y_N$.

On the basis of the above setting, the SLAM problem of robot vision can be expressed as the following mathematical model:

a)Motion model

$$x_k = f(x_{k-1}, u_k, w_k) \quad (7)$$

Among them, u_k is the reading of the motion sensor and w_k is the noise. Function f is used to describe the process which does not specify its way of action, that is, this function can refer to any motion sensor, such as encoder or inertial sensor[8].

b)Observation model

The observation equation[9] is described as, when the robot looks at the point of a path punctuation y_j in the position x_k, it produces an observation data $z_{k,j}$. So, you can use a function to show it

$$z_{k,j} = h(y_j, x_k, v_{k,j}) \tag{8}$$

Among them, $v_{k,j}$ is the noise in this observation. The function h is the same as f, and does not specify the way of action.

After these two models, we parameterize the state of the robot according to the types of the sensors we use:

1)The robot moves in space, then its position can be described by three positions and three attitude angles, that is $x_k = [x, y, z, \phi, \varphi, \gamma]_k^T$. At the same time, the motion sensor can measure the change of the position and attitude angle of the robot at any two time intervals $u_k = [\Delta x, \Delta y, \Delta z, \Delta\phi, \Delta\varphi, \Delta\gamma]_k^T$, so the motion equation can be parameterized:

$$\begin{bmatrix} x \\ y \\ z \\ \phi \\ \varphi \\ \gamma \end{bmatrix}_k = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \varphi \\ \gamma \end{bmatrix}_{k-1} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\phi \\ \Delta\varphi \\ \Delta\gamma \end{bmatrix}_k + w_k \tag{9}$$

Among them, the pitch angle, the roll angle and the yaw angle are respectively expressed by ϕ, φ, γ.

2)When moving a robot to observe a 3D path punctuation point through a camera the robot can get two quantities: the distance r between the punctuation point and the robot body and the angle θ between them. To mark the point of the road, the observation data is $z = [r, \theta]^T$, then the observation equation is parameterized:

$$\begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{(p_x - x)^2 + (p_y - y)^2} \\ \arctan(\dfrac{p_y - y}{p_x - x}) \end{bmatrix} \tag{10}$$

The above (7), (8), (9), (10) formulas describe the basic visual SLAM problem.

4.3 Monocular vision SLAM

1)The processing of the image obtained by the camera

We use the SURF method to detect and extract the feature points collected by the single camera, so that the features have rotation invariance and scale invariance. Because of the real-time requirement, we use fast approximation nearest neighbor (FLANN) algorithm to match the feature points and add Hamming distance constraints to match the number of constraints and filter and get the correct matching.

2)Estimate the motion of the camera and obtain the local map

The information we can get from a monocular camera is 2D pixel coordinates. Therefore, we need to estimate the motion of cameras based on two sets of 2D points, and use the product geometric method to solve them on the premise of correct feature matching. The basic matrix E and the essential matrix F are obtained according to the pixel position of the matching points, and then the two ones are used to solve R、t. Then the position of the camera can be expressed as:

$$x_k = Rx_{k-1} + t \tag{11}$$

Among them, for the K amplitude diagram, R、t for the transformation matrix for the K-1 amplitude map to the K amplitude diagram.

At the same time, we directly lead to the scale of monocular vision uncertainty as a result of the length normalization of t, so we need to have an initialization procedure in SLAM, that is, two images of the initial translation must have a certain extent, the later trajectories and maps are based on this translation.

3)Optimize the position and local map that have been obtained

We consider all the motions and observations, which form a least square problem. When we only observe the SLAM of the equation, we use the nonlinear optimization method to select the frames with common observations as the key frames to solve the BA, and pose and optimize the graph, so as to get the globally consistent trajectories and maps.

4)Loop detection

In order to ensure the correctness of our estimated trajectories and maps in long time, we use loop detection[10] to eliminate accumulated errors in the SLAM process based on the correlation between the current data and historical data. This part is mainly divided into two processes, which are loop detection and loop correction. The loop detection first uses the BOW to detect the dictionary, and then uses the DBoW3 library to calculate the similarity transformation. Loop correction is mainly the fusion of loop detection and the graph optimization of key frames.

5)Mapping

Maps are drawn for location, so we can draw different types of maps according to the requirements. Generally divided into two

types: the first is a sparse map map service to locate, only interested in the modeling part, is also characteristic; the second is a dense map service in navigation, obstacle avoidance or 3D reconstruction, modeling all seen, can clearly determine what parts of the map can pass where not through. This article draws a sparse map that serves the location of the map.

5. Analysis of experiment

5.1 Experimental platform

The experiment platform is as follows:

1)the experiment host computer is Acer aspire V5-471G notebook, its operating system is Ubuntu14.04 system.

2)the experiment monocular camera is the HDwebcam of the notebook.

3)the experimental platform is the ROS system.

4)the experimental mobile platform is a four round omnidirectional mobile robot CA-OMinar.

The following figure3 are the experimental platform for the experiment:

Figure 3 experimental platform

The experiment uses two ways to verify the correctness and feasibility of this method:

Mode one: using the KITTI dataset to run the data set on this platform to analyze the positioning effect and the composition effect of the experimental method.

Mode two: using a notebook single camera to collect real-time data to analyze the effectiveness and existing problems of the experimental method.

5.2 Experimental result

Mode one experimental results:

Figure 4 some pictures of dataset

We use part of the KITTI dataset for off-line operation. The data set consists of 153 grayscale images and some pictures some of them as shown figure 4.

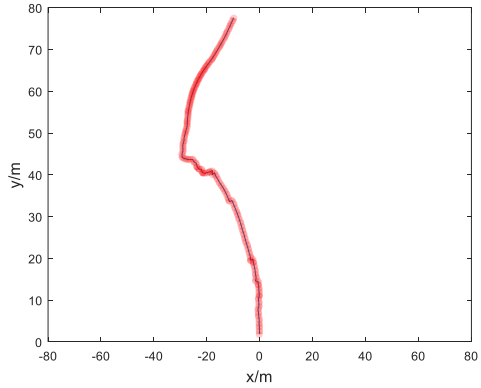

Figure 5 calculated trajectory

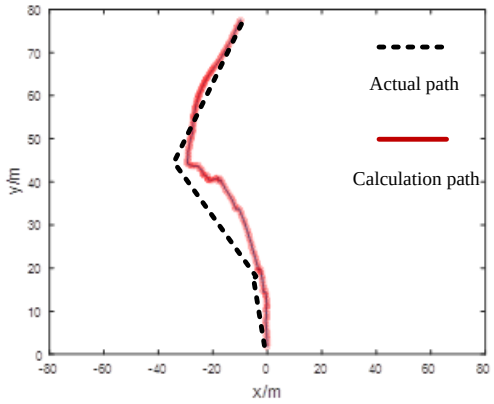

Figure 6 compares of two trajectories

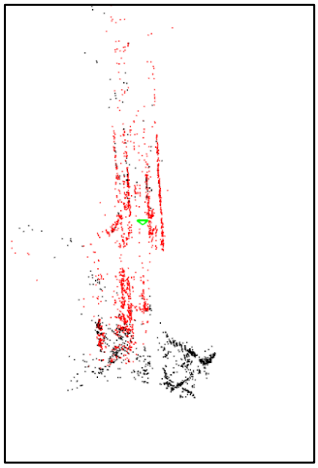

Figure 7 constructed sparse map

The experimental results are shown in Figure 5-8. Figure 5 the camera running trajectory calculated by monocular SLAM system. Figure 6 the comparison between the calculated trajectory and the actual trajectory. Figure 7 a sparse map constructed by monocular SLAM system, and figure 8 the trajectory of the camera in the sparse map.

Mode two experimental results.

We use a notebook camera to collect image data directly and deal with it in real time. Figure 9 are the environment. Figure 10 are

the feature capture of camera on the environment. Figure 11 Environmental sparse map built by camera. Figure 12 a sparse map constructed by monocular SLAM system. Figure 13 the comparison between the calculated trajectory and the actual trajectory.

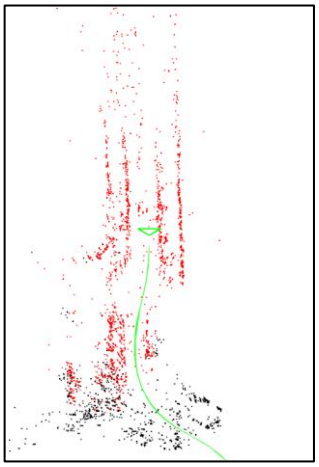

Figure 8 the trajectory of the camera in map

Figure 9 the environment

Figure 10 feature capture of camera

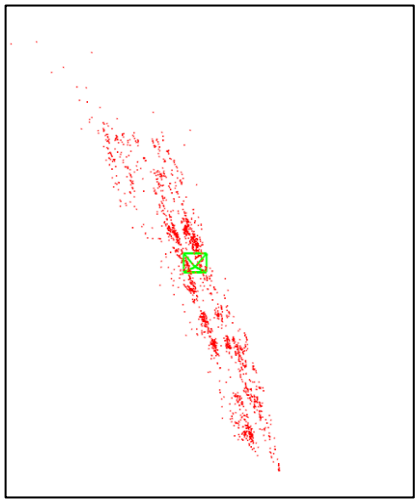

Figure 11 constructed sparse map

5.3 Analysis of experimental results

According to the above two ways of experimental results, we can see: in positioning, two modes of positioning effect is ideal, and in composition, the effect of running on dataset is better than the actual operation effect.

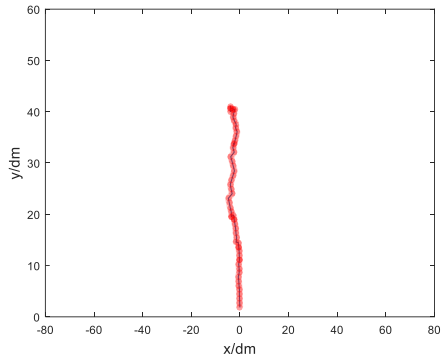

Figure 12 calculated trajectory

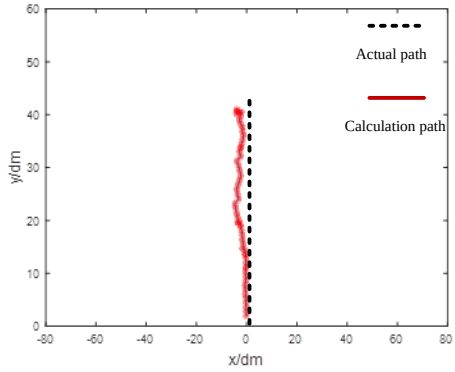

Figure 13 compares of two trajectories

6. Conclusion

The main reason why the mapping effect of running on dataset is better than the actual operation effect is that the dataset of the images are through processing and with relatively good performance of camera and camera pixels, but performance of camera and camera pixels of ours are very low, so the feature point extraction is obviously less than the dataset, the composition effect is no better than dataset.

Acknowledgements

Authors would like to acknowledge technical and financial support provided by Lei Cheng.

References

Y. Cheng, J. Bai and C. Xiu, "Improved RGB-D vision SLAM algorithm for mobile robot,"2017 29th Chinese Control And Decision Conference, Chongqing, 2017, pp. 5419-5423.

K. N. Al-Mutib, E. A. Mattar, M. M. Alsulaiman and H. Ramdane, "Stereo vision SLAM based indoor autonomous mobile robot navigation,"2014 IEEE International Conference on Robotics and Biomimetics, Bali, 2014, pp. 1584-1589.

Y. You, "The Research of SLAM Monocular Vision Based on The Improved SURF Feather,"2014 International Conference on Computational Intelligence and Communication Networks, Bhopal, 2014, pp. 344-348.

J. Liu, H. Chen and B. Zhang, "Square Root Unscented Kalman Filter based ceiling vision SLAM,"2013 IEEE International Conference on Robotics and Biomimetics, Shenzhen, 2013, pp. 1635-1640.

R. Schattschneider, G. Maurino and Wenhui Wang, "Towards stereo vision SLAM based pose estimation for ship hull inspection,"OCEANS' MTS/IEEE KONA, Waikoloa, HI, 2011, pp. 1-8.

Azizi A, Nourisola H, Ghiasi A. R. 3D inertial algorithm of SLAM for using on UAV[C] // 2016 4th International Conference on Robotics and Mechatronics, Tehran, 2016: 122-129.

Y. T. Wang and Y. C. Fang, "Robust mono-vision SLAM of mobile robots,"Proceedings of the 30th Chinese Control Conference, Yantai, 2011, pp. 3953-3957.

X. H. Wang and P. f. Li, "Improved Data Association Method in Binocular Vision-SLAM,"2010 International Conference on Intelligent Computation Technology and Automation, Changsha, 2010, pp. 502-505.

E. Wu, L. Zhao, Y. Guo, W. Zhou and Q. Wang, "Monocular vision SLAM based on key feature points selection,"The 2010 IEEE International Conference on Information and Automation, Harbin, 2010, pp. 1741-1745.

D. X. Zhu, "Binocular Vision-SLAM Using Improved SIFT Algorithm,"2010 2nd International Workshop on Intelligent Systems and Applications, Wuhan, 2010, pp. 1-4.

Rui Peng was born in Ganghuang Hubei Province in 1993. He is a master degree candidate at Wuhan University of Science and Technology.His research direction is the positioning and navigation of the mobile robot.

Lei Cheng is a Professor in Control Science and Engineering at Engineering Research Center of Metallurgical Automation and Measurement Technology, Ministry of Education, Wuhan University of Science and Technology, Wuhan, China. He obtained his Bachelor, Master and PhD Degree in Huazhong University of Science and Technology, China, in 1999, 2002 and 2005. His research interests include robot and machine perception, embedded system design, Internet of things technology and intelligent control.

Yating Dai was born in Yingchang Hubei Province in 1992. She is a master degree candidate at Wuhan University of Science and Technology. Her research interests include computer vision and vision-aided navigation.

Xitong Zhao was born in Harbin in 1994. She has studying the M.S. degrees in Control engineering from the Wuhan University of science and technology. Her main research interest include robot vision navigation.

Huaiyu Wu is a member of the Teaching Guidance Committee of the Automation Department of the Ministry of education, the senior member of the American Institute of electrical and Electronic Engineers (SMIEEE) and the vice president of Hubei automation society. He graduated from Tsinghua University. His research interests include computer control and application; digital image and machine vision detection; electromechanical system integration and control of service robot and its control.

Yang Chen is a member of the China Society of automation, the director of the Hubei artificial intelligence society. He graduated from the State Key Laboratory of robotics, Institute of automation, Shenyang Institute of automation, Chinese Academy of Sciences, and received a doctorate in engineering. The main research areas: mobile robot modeling, planning and control, machine learning and human-computer interaction.