Improved Stereo Vision Robot Locating and Mapping Method

Yu Haige
School of computer science and engineering
Xi'an Technological University
Xi'an, Shaanxi, China
E-mail: 279084342@qq.com

Wei Yanxi
School of computer science and engineering
Xi'an Technological University
Xi'an, Shaanxi, China
E-mail: 407171251@qq.com

Yu Fan
School of computer science and engineering
Xi'an Technological University
Xi'an, Shaanxi, China
E-mail: yffshun@163.com

***Abstract*—Vision-based SLAM has an outstanding problem is not work when the camera fast motion, or camera operating environment characterized by scarce. Aiming at this problem, this paper proposes a SLAM method of IMU and vision fusion. This article uses a stereo camera to extract the image ORB feature points. During the camera movement, if the number of extracted feature points is less than a certain threshold and the camera movement cannot be estimated or the estimated camera rotation and translation is greater than a certain threshold, the camera pose is estimated by fusing IMU , Otherwise use feature points to estimate camera pose. This paper uses non-linear optimization methods to perform pose estimation of pure feature points and pose estimation of fused IMU, respectively. The experimental results show that binocular vision SLAM with IMU information can estimate the camera pose more accurately.**

Keyword-Robot; IMU; Stereo Vision; SLAM

I. INTRODUCTION

With the development of robot technology, more and more robots are approaching our lives, such as sweeping robots, shopping mall robots, etc. Mobile robots are the product of the cross fusion of various disciplines and technologies. Among them, SLAM(Simultaneous Localization and Mapping) is an important technology for mobile robots. SLAM means that the robot builds a map of the surrounding environment in real time based on sensor data without any prior knowledge, and infers its own positioning based on the map. From the 1980s to the present, more and more sensors are used in SLAM, from early sonar, to later 2D/3D lidar, to monocular, binocular, RGBD, ToF and other cameras. Compared with lidar, cameras used in vision SLAM have become the focus of current SLAM research due to their advantages such as low price, light weight, large amount of image information, and wide application range. Stereo cameras generally consist of two pinhole cameras placed horizontally. Compared to monocular vision's scale uncertainty and pure rotation problems, binocular cameras can directly calculate the pixel depth. At the same time, compared to RGB-D cameras, stereo cameras collect images

DOI: 10.21307/ijanmc-2019-070

directly from ambient light and can be used indoors and outdoors. Compared with lidar, the main disadvantage of the camera as a SLAM sensor is that when the camera moves too fast, the camera will blur images, and the camera will not work in a scene with insufficient environmental feature textures and few feature points.

Aiming at the problems of the above-mentioned visual SLAM system, this paper proposes an algorithm that fuses IMU and SLAM. Through the fusion of IMU, it can provide a good initial pose for the system. At the same time, during the camera movement process, it makes up for the shortcomings of visual SLAM, ensuring the accuracy of the camera pose estimation in the case of fast camera movement and lack of environmental texture.

II. RELATED WORKS

A. Camera coordinate system

Camera models generally have four coordinate systems: a pixel coordinate system, an image coordinate system, a world coordinate system, and a camera coordinate system. Figure 1:

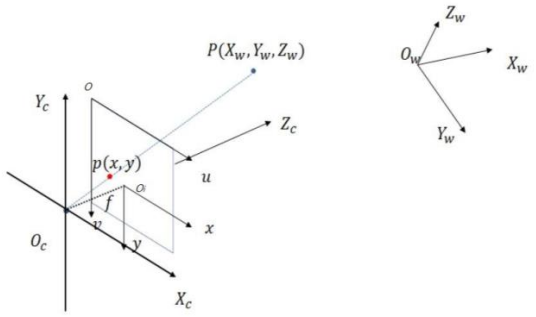

Figure 1. Camera related coordinate system

Among them, $O_w - X_w Y_w Z_w$ is the world coordinate system. The world coordinate system is the reference coordinate system in the visual SLAM system. The positions of the camera trajectory and map points are described based on this coordinate system. The unit is m.

$O_i - xy$ is the image coordinate system. The image coordinate system uses the intersection of the camera optical center and the image plane coordinate system as the origin. The unit is mm.

$O_c - X_c Y_c Z_c$ is the camera coordinate system. The camera coordinate system uses the camera optical center as the origin, and the directions parallel to the x-axis and y-axis of the image coordinate system are respectively taken as the X_c-axis and Y_c-axis, and the direction perpendicular to the image plane is the Z_c-axis. The unit is m.

$O-uv$ is the pixel coordinate system. The origin of the pixel coordinate system is generally the upper left corner of the image, with the u axis to the right parallel to the x axis, and the v axis to the y axis. The unit is pixel.

B. Camera projection model

The camera maps the coordinate points of the three-dimensional world to the two-dimensional image plane. This process is generally a pinhole model. Under the pinhole model, it is assumed that there is a spatial point P, and the coordinates of the point P are $[X, Y, Z]^T$. After the projection of the small hole O, the point P falls on the imaging plane $o-xy$, and the imaging point is p, The p-point coordinate is $[x, y, z]^T$. Let the distance from the imaging plane to the small hole be the focal length f. Therefore, according to the principle of triangle similarity, there are:

$$\frac{Z}{f} = \frac{X}{x} = \frac{Y}{y} \tag{1}$$

So we can get:

$$\begin{cases} x = f\dfrac{X}{Z} \\ y = f\dfrac{Y}{Z} \end{cases} \tag{2}$$

The difference between the pixel coordinate system and the imaging plane is a zoom and a translation of the origin. Suppose that the pixel coordinates are scaled α times on the u axis and β times on the v axis, and the origin is translated $\left[c_x, c_y\right]^T$, so we can get:

$$\begin{cases} u = \alpha x + c_x \\ v = \beta y + c_y \end{cases} \tag{3}$$

Equation (3) is substituted into equation (2) to get:

$$\begin{cases} u = f_x\dfrac{X}{Z} + c_x \\ v = f_y\dfrac{Y}{Z} + c_y \end{cases} \tag{4}$$

The unit of f is m and the unit of α and β is $pixel/m$, so the unit of f_x, and f_y is $pixel$. Written as a matrix:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \triangleq \frac{1}{Z}\mathbf{KP} \tag{5}$$

Among them, the matrix K is called the internal parameter matrix of the camera, and P is the coordinate representation of the space point in the camera coordinate system.

Let the coordinate P of the space point in the camera coordinate system correspond to the coordinate P_w in the world coordinate system, and use coordinate transformation to obtain:

$$Z\mathbf{P}_{uv} = Z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}(\mathbf{R}\mathbf{P}_w + \mathbf{t}) = \mathbf{K}\mathbf{T}\mathbf{P}_w \tag{6}$$

Among them, T represents the pose of the camera relative to the world coordinate system, and can also be called the external parameter of the camera. In summary, the pinhole camera model uses the triangle similarity relationship to obtain the relationship between space points and pixels, which is a relatively ideal model. In practice, there will be errors in the manufacture and installation of optical lenses, which will affect the propagation of light during the imaging process and cause distortion in the images collected by the camera. Here we mainly consider radial distortion and tangential distortion.

Radial distortion is caused by the shape of the lens, and the distortion increases as the distance between the pixel and the center of the image increases. Therefore, a polynomial function can be used to describe the changes before and after the distortion, that is, the quadratic and higher-order polynomial functions related to the distance between the pixel and the center of the image can be used for correction. The polynomial of the coordinate change before and after the radial distortion correction is as follows:

$$\begin{cases} x_{corrected} = x(1 + k_1 \mathrm{r}^2 + k_2 \mathrm{r}^4 + k_3 \mathrm{r}^6) \\ y_{corrected} = y(1 + k_1 \mathrm{r}^2 + k_2 \mathrm{r}^4 + k_3 \mathrm{r}^6) \end{cases} \tag{7}$$

Among them, $\left[x, y\right]^T$ is the coordinates of the uncorrected points, and $\left[x_{corrected},\ y_{corrected}\right]^T$ is the coordinates of the points after the distortion is corrected. r is the distance from the point (x, y) to the origin. k_1, k_2 and k_3 are three radial distortion

parameters. Usually these three parameters can be obtained by the calibration step.

For tangential distortion, the reason is that the lens and the imaging plane cannot be strictly parallel during camera assembly. Tangential distortion can be corrected using two other parameters, p1 and p2:

$$\begin{cases} x_{corrected} = x + 2p_1xy + p_2(\mathrm{r}^2 + 2x^2) \\ y_{corrected} = y + 2p_2xy + p_1(\mathrm{r}^2 + 2y^2) \end{cases} \quad (8)$$

Considering the two types of distortion, we can find the correct position of a pixel in the pixel coordinate system through 5 distortion coefficients:

$$\begin{cases} x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1xy + p_2(r^2 + 2x^2) \\ y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_2xy + p_1(r^2 + 2y^2) \end{cases} \quad (9)$$

In summary, the parameters describing the camera model mainly include: in the camera's internal parameter matrix, and distortion correction parameters.

C. Stereo camera ranging principle

The binocular camera generally consists of two pinhole cameras placed horizontally, and the two cameras observe an object together. The aperture centers of both cameras are located on one axis, and the distance between the two is called the baseline b of the binocular camera. There is an existing space point P, which is an image in the left-eye camera and the right-eye camera, and is denoted as P_L, P_R. Due to the presence of the camera baseline, these two imaging positions are different. Remember that the coordinates of the imaging on the left and right sides are x_L, x_R, which can be seen from the similarity of the triangles:

$$\frac{z - f}{z} = \frac{b - u_L + u_R}{b} \quad (10)$$

We can get:

$$z = \frac{fb}{d} \quad (11)$$

The above model is an ideal model, which aims to explain the principle of measuring the actual three-dimensional point depth of the binocular camera. In practical applications, due to factors such as manufacturing and installation, it is difficult to achieve that the imaging planes of the binocular cameras are strictly on the same plane and the optical axes are strictly parallel. Therefore, before using a binocular camera for measurement, it should be calibrated to obtain the left and right camera internal parameters and the relative position relationship between the left and right cameras.

III. POSE ESTIMATION ALGORITHM

At present, the fusion method of monocular vision sensor and IMU can be divided into two types: loose coupling and tight coupling[1]. Loose coupling is based on the vision sensor and IMU as two separate modules, both of which can calculate the pose information, and then fused by EKF[2] and so on. Tight coupling refers to the non-linear optimization of vision and IMU data to obtain pose estimates. Because tight coupling can make full use of each sensor's data, this paper uses tight coupling to fuse vision and IMU data. Firstly, the purely visual feature point pose estimation method is used to estimate the camera pose. Then, during the camera movement, if the number of extracted feature points is less than a certain threshold value, the camera movement cannot be estimated or the estimated camera rotation and translation are greater than a certain threshold value, The camera pose is estimated by fusing the IMU, otherwise feature points are still used to estimate the camera pose.

A. Pose estimation using pure visual information

The ORB (Oriented Fast and rotated Brief) algorithm was proposed by Ethan Rublee et al. In 2011[3]. The ORB feature is composed of the FAST

feature and the BRIEF descriptor. It adds orientation and scale invariance to the FAST feature. Features are described using binary BRIEF descriptors. When performing feature matching, the descriptors between feature points and feature points are compared. The binocular camera can directly obtain the corresponding 3D position of the pixel under the known pixel matching of the left and right camera images. Therefore, the stereo camera-based SLAM system can use the known 3D point and its projection match in the current frame to obtain the current camera pose without the need to solve camera motion using epipolar geometry[4].

This paper first uses the method of EPnP[5] to solve the camera pose. The EPnP pose solution method can more effectively use the matching point information, and iteratively optimize the camera pose. EPnP is known as the coordinates $\left\{P_i^w, i=1,2,...,n\right\}$ of n space points in the world coordinate system and their corresponding coordinates $\left\{P_i^c, i=1,2,...,n\right\}$ in the image coordinate system to solve the rotation matrix R and translation vector t of the camera movement.Set four non-coplanar virtual control points in the world coordinate system, whose homogeneous sitting marks are: $\left\{C_i^w \mid i=1,2,3,4\right\}$ .The relationship between the world coordinates of the space points and the control points is as follows:

$$P_i^w = \sum_{j=1}^{4} \alpha_{ij} C_j^w, with \sum_{j=1}^{4} = \alpha_{ij} = 1 \tag{12}$$

Once thevirtual control point is determined and the premise that the four control points are not coplanar, $\left\{\alpha_{ij}, j=1,...,4\right\}$ is the only one determined.In the image coordinate system, the same weighting sum relationship exists:

$$P_i^c = \sum_{j=1}^{4} \alpha_{ij} C_j^c$$

Substituting equation (13) into the camera model gives:

$$\forall i, s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{K} P_i^c = \mathbf{K} \sum_{j=1}^{4} \alpha C_j^c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^{4} \alpha_{ij} \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix} \tag{13}$$

The image coordinates u_i, v_i in Equation (13) are known, so:

$$s_i = \sum_{j=1}^{4} \alpha_{ij} z_j^c \tag{14}$$

From equations (13) and (14):

$$\begin{cases} \sum_{j=1}^{4} \alpha_{ij} f_x x_j^c + \alpha_{ij} (c_x - u_i) z_j^c = 0 \\ \sum_{j=1}^{4} \alpha_{ij} f_y y_j^c + \alpha_{ij} (c_y - v_i) z_j^c = 0 \end{cases} \tag{15}$$

In order to obtain the coordinates of the 2D point into the camera coordinate system, it is assumed that α_{ij} in the camera coordinate system is consistent with α_{ij} in the world coordinate system, that is, to find the rotation and translation of the four control points. Solve linear equations:

$$\mathbf{M}\mathbf{X} = 0 \tag{16}$$

Among them, M is a $2n \times 12$ matrix, and $\mathbf{X} = [C_1^{cT}, C_2^{cT}, C_3^{cT}, C_4^{cT}]$ is a vector composed of 12 unknowns to be solved.

$$\mathbf{X} = \sum_{i=1}^{N} \beta_i v_i \tag{17}$$

v_i is the right singular vector of M, and the corresponding singular value is 0. Solve the $\mathbf{M}^T\mathbf{M}$ eigen value and eigenvector. The eigenvector with eigenvalue of 0 is v_i. N is the dimension of the $\mathbf{M}^T\mathbf{M}$ space, and β_i is the coefficient to be determined.

Depending on the position of the reference point, the spatial dimension of the matrix $\mathbf{M}^T\mathbf{M}$ may take the values 1,2,3,4. According to the same distance between the control points in the world coordinate system and the camera coordinate system, six constraints can be obtained, and the pending coefficients can be solved.

When N = 1, according to the constraints:

$$\left\|\beta v^{[i]} - \beta v^{[j]}\right\|^2 = \left\|C_i^w - C_j^w\right\|^2 \tag{18}$$

and so：

$$\beta = \frac{\sum_{[i,j]\in[1,4]} \left\|v^{[i]} - v^{[j]}\right\| \bullet \left\|C_i^w - C_j^w\right\|}{\sum_{[i,j]\in[1,4]} \left\|v^{[i]} - v^{[j]}\right\|^2} \tag{19}$$

When N = 2:

$$\left\|\beta_1 v_1^{[i]} + \beta_2 v_2^{[i]} - (\beta_1 v_1^{[j]} + \beta_2 v_2^{[j]})\right\|^2 = \left\|C_i^w - C_j^w\right\|^2 \tag{20}$$

Since β_1 and β_2 only appear in the equation as quadratic terms, let $\boldsymbol{\beta} = [\beta_1^2, \beta_1\beta_2, \beta_2^2]^T$, and use the vector $\boldsymbol{\rho}$ to represent all $\left\|C_i^w - C_j^w\right\|^2$, thus obtaining the equation:

$$\mathbf{L}\boldsymbol{\beta} = \boldsymbol{\rho} \tag{21}$$

Where $\mathbf{L}$ is a 6×3 matrix composed of v_1 and v_2.

When N = 3, $\mathbf{L}$ is a 6×6 matrix.

In summary, the coordinate solution of the reference point in the camera coordinate system can be obtained as the initial value of the optimization, the optimization variable is $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_N]^T$, and the objective function is:

$$Error(\boldsymbol{\beta}) = \sum_{(i,j)s.t.i<j} (\left\|C_i^c - C_j^c\right\|^2 - \left\|C_i^w - C_j^w\right\|^2) \tag{22}$$

Optimize $\boldsymbol{\beta}$ corresponding to the smallest dimension of the error, get the vector $\mathbf{X}$, and restore the coordinates of the control point in the camera coordinate system. At the same time, the coordinates of the reference point in the camera coordinate system are obtained according to the centroid coordinate coefficient. Finally, according to the coordinates of a set of point clouds in the two coordinate systems, the pose transformations of the two coordinate systems are obtained. The solution steps are as follows:

a) Find the center point:

$$p_c^c = \frac{\sum p_i^c}{n}, P_c^w = \frac{\sum p_i^w}{n} \tag{23}$$

b) To the center point:

$$q_i^c = p_i^c - p_c^c, q_i^w = p_i^w - p_c^w \tag{24}$$

c) Calculate the H matrix:

$$\mathbf{H} = \sum_{i=1}^{n} q_i^c q_i^{w^T} \tag{25}$$

d) SVD decomposition of H matrix:

$$\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^T \tag{26}$$

e) Calculate the rotation R:

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T \tag{27}$$

If R <0, then R(2,.) =-R(2,0).

f) Calculate displacement t:

$$t = p_0^c - \mathbf{R} p_0^w \tag{28}$$

Taking the results of EPnP solution as initial values, the method of g2o was used to optimize the pose of the camera nonlinearly. Construct the least squares problem and find the best camera pose:

$$\xi^* = \arg\min_{\xi} \frac{1}{2} \sum_{i=1}^{n} \left\| u_i - \frac{1}{s_i} K \exp(\xi^\wedge) P_i \right\|_2^2 \tag{29}$$

Among them, u_i is the pixel coordinates of the projection point, K is the camera internal reference, ξ is the camera pose, and P_i is the space point coordinate.

B. Camera pose estimation method based on IMU

The measurement frequency of the IMU is often higher than the frequency at which the camera collects pictures. For example, the binocular camera used in this article has a frame rate of up to 60FPS and an IMU frequency of up to 500Hz. The difference in frequency between the two results in multiple IMU measurements between the two frames. Therefore, in order to ensure the information fusion of the two sensors, it is necessary to pre-integrate [6] the data of the IMU. That is, only the IMU information between the two image moments is integrated to obtain the relative pose value, and the integration result is saved for later joint optimization.The IMU-based camera pose estimation method mainly includes three coordinate systems: the world coordinate system, the IMU coordinate system, and the camera coordinate system. All pose and feature point coordinates are finally expressed in the world coordinate system. During the calculation process, this article will convert the state quantity in the camera coordinate system to the IMU coordinate system, and then to the world coordinate system.In this article, the letter W is used to represent the world coordinate system, the letter B is used to represent the IMU coordinate system, R_{WB} is used to represent the rotation matrix from the IMU coordinate system to the world coordinate system, and p_{WB} is used to represent the translation matrix from the IMU coordinate system to the world coordinate system.

The acceleration and angular velocity of the IMU are:

$$\begin{aligned} {}_B\tilde{\omega}_{WB}(t) &= {}_B\omega_{WB}(t) + b^g(t) + \eta^g(t) \\ {}_B\tilde{\mathrm{a}}_{WB}(t) &= R_{WB}^T(t)({}_W\mathrm{a}(t) - {}_W g) + b^a(t) + \eta^a(t) \end{aligned} \tag{30}$$

Among them, $b^a(t)$ and $b^g(t)$ represent the bias of the accelerometer and gyroscope respectively, $\eta^a(t)$ and $\eta^g(t)$ represent the noise of the accelerometer and gyroscope respectively, and ${}_W g$ represents the gravity vector in the world coordinate system.

The derivatives of rotation, velocity, and translation are expressed as:

$$\begin{aligned} \dot{R}_{WB} &= R_{WB}\,{}_B\omega_{WB}^\wedge \\ {}_W\dot{v}_{WB} &= {}_W a_{WB} \\ {}_W\dot{p}_{WB} &= {}_W v_{WB} \end{aligned} \tag{31}$$

The rotation, speed and translation in the world coordinate system can be obtained by the general integral formula:

$$\begin{aligned} R_{WB}(t+\Delta t) &= R_{WB}(t) Exp\left(\int_t^{t+\Delta t} {}_B\omega_{WB}(\tau) d\tau\right) \\ {}_W v(t+\Delta t) &= {}_W v(t) + \int_t^{t+\Delta t} {}_W\mathrm{a}(\tau) d\tau \\ {}_W p(t+\Delta t) &= {}_W p(t) + \int_t^{t+\Delta t} {}_W v(\tau) d\tau + \iint_t^{t+\Delta t} {}_W\mathrm{a}(\tau) d\tau^2 \end{aligned} \tag{32}$$

Use Equation (32) in discrete time for Euler integration:

$$\begin{aligned} R_{WB}(t+\Delta t) &= R_{WB}(t)Exp({}_B\omega_{WB}(t)\Delta t) \\ {}_W v(t+\Delta t) &= {}_W v(t) + {}_W \mathrm{a}(t)\Delta t \\ {}_W p(t+\Delta t) &= {}_W p(t) + {}_W v(t)\Delta t + \frac{1}{2}{}_W \mathrm{a}(t)\Delta t^2 \end{aligned} \tag{33}$$

The IMU model is obtained from equations (30) and (33):

$$\begin{aligned} R(t+\Delta t) &= R(t)Exp((\tilde{\omega}(t) - b^g(t) - \eta^{gd}(t)\Delta t) \\ v(t+\Delta t) &= v(t) + g\Delta t + R(t)(\tilde{\mathrm{a}}(t) - b^a(t) - \eta^{ad}(t))\Delta t \\ p(t+\Delta t) &= p(t) + v(t)\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}R(t)(\tilde{\mathrm{a}}(t) - b^a(t) - \eta^{ad}(t))\Delta t^2 \end{aligned}$$

Suppose there are two image frames with time t_i and t_j, $t_j > t_i$. Therefore, the IMU's pre-integration observation model is:

$$\begin{aligned} \Delta\tilde{R}_{ij} &= R_i^T R_j Exp(\delta\phi_{ij}) \\ \Delta\tilde{v}_{ij} &= R_i^T(v_j - v_i - g\Delta t_{ij}) + \delta v_{ij} \\ \Delta p_{ij} &= R_i^T(p_j - p_i - v_i\Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2) + \delta p_{ij} \end{aligned} \tag{34}$$

Among them, A, B, and C are the noise terms of the rotation amount, the pre-integrated speed noise term, and the pre-integrated translation noise term, respectively.

For the pose between two adjacent frames, this paper still uses a nonlinear optimization method to fuse IMU information and visual information. Among them, the state quantities that need to be optimized are:

$$\theta = \left\{R_{WB}^j, {}_W p_B^j, {}_W v_B^j, b_g^j, b_a^j\right\} \tag{35}$$

In equation (36), R_{WB}^j, v_{WB}^j, and p_{WB}^j are the rotation, velocity, and translation of the IMU coordinate system relative to the world coordinate system at time i, and the random walk bias of the gyroscope and accelerometer at time i, respectively.

Therefore, the optimal state quantity θ is solved by optimizing the visual reprojection error and the IMU measurement error:

$$\theta^* = \arg\min_{\theta}(\sum_k E_{proj(k,j)} + E_{IMU}(i,j)) \tag{36}$$

C. Experimental design

The upper computer of the experimental platform in this article is a laptop with Ubuntu 16.04 version, running memory is 8G, processor model is CORE i5 8250U, and the main frequency is 1.6GHz. The robot platform is a Dashgo D1 robot mobile platform that supports the ROS development system. The overall size is $\Phi 406\times 210$ and the diameter of the driving wheel is 125mm. The binocular camera sensor used is MYNT EYE D1000-IR-120/Color.

The experiments in this paper are mainly aimed at the positioning accuracy of the robot. The evaluation index is the RMSE (root-mean-square-error) of the robot position:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - p_i)^2} \tag{37}$$

Where $\hat{p}_i$ is the estimated robot position and p_i is the actual robot position.

Figure 2. Robot Straight Driving Positioning Experiment

In this paper, robot positioning experiments are performed in corridor environments with insignificant environmental characteristics and indoor environments with rich characteristics. In a corridor environment, a mobile robot is used to carry experimental equipment to travel at a constant speed of 10m in the positive direction of the camera, and then the positioning accuracy of pure vision and the positioning accuracy of vision fusion IMU are recorded separately. In a feature-rich indoor environment, a robot linear experiment was also performed to make the mobile robot move forward at a constant speed of 5m in the positive direction of the camera, but the speed was 2.5 times that of the previous experiment. Perform multiple experiments and record the results.

TABLE I. EXPERIMENTAL RESULT

Robot operating environment	Pure visual RMSE/m	Visual fusion IMU RMSE/m
Low-texture corridor environment	0.0746	0.02122
Feature-rich environment	0.1024	0.06502

From the experimental results, it can be seen that the stereo vision positioning error of the fusion IMU is less than the pure vision positioning error, which indicates that the visual positioning of the robot with the fusion IMU is more accurate than the vision-only positioning in low-texture environments and fast robot movements. degree.

IV. CONCLUSION

In this paper, the robot positioning technology in the robot system is researched, and a binocular vision fusion IMU-based robot positioning method is proposed. Compared with the pure vision robot localization method, the proposed method is more robust in low-textured environments and fast robot movements. The experimental results show that the visual positioning method integrated with IMU solves the defects of pure visual positioning to a certain extent and improves the positioning accuracy of the robot.

REFERENCE

[1] Agostino Martinelli. Closed-Form Solution of Visual-Inertial Structure from Motion[J]. International Journal of Computer Vision, 2014, 106(2):138-152.

[2] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. International Journal of Robotics Research, 1986, 5(4): 56-68.

[3] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]// 2011 International Conference on Computer Vision. IEEE, 2012.

[4] Gao Xiang, Zhang Tao. Fourteen lectures on visual SLAM [M]. Beijing: Publishing House of Electronics Industry, 2017.

[5] V. Lepetit, F. Moreno-Noguer, P. Fua. EPnP:An accurate o(n) solution to the pnp problem[J]. International Journal of Computer Vision,2008,81(2):155-166.

[6] Forster C, Carlone L, Dellaert F, et al. On-Manifold Preintegration for Real Time Visual--Inertial Odometry[J]. IEEE Transactions on Robotics, 2017, 33(1):1-21.