CrossMark

Motion estimation of indoor robot based on image sequences and improved particle filter

Xiaoming Dong[1] · **Liefu Ai**[1] · **Rong Jiang**[2]

Abstract Robot motion estimation is fundamental in most robot applications such as robot navigation, which is an indispensable part of future internet of things. Indoor robot motion estimation is difficult to be resolved because GPS (Global Positioning System) is unavailable. Vision sensors can provide larger amount of image sequences information compared with other traditional sensors, but it is subject to the changes of light. In order to improve the robustness of indoor robot motion estimation, an enhanced particle filter framework is constructed: firstly, motion estimation was implemented based on the distinguished indoor feature points; secondly, particle filter method was utilized and the least square curve fitting was inserted into the particle resampling process to solve the problem of particle depletion. The various experiments based on real robots show that the proposed method can reduce the estimation errors greatly and provide an effective resolution for the indoor robot localization and motion estimation.

1 Introduction

Mobile robots' self-localization and position tracking is one of the most important tasks in various applications such as visual navigation and visual servo, which are the indispensable terminal applications for the future Internet of Things. A common way to implement this task is to estimate robots' pose according to inertial sensors. This method is difficult to resolve the problem of robot's wheel slipping, so the

---

✉ Rong Jiang
  jiang_rong@aliyun.com

[1]   The University Key Laboratory of Intelligent Perception and Computing of Anhui Province, Anqing
     Normal University, Anqing 246133 Anhui, China

[2]   School of Information, Yunnan University of Finance and Economics, Kunming 650221, China

🖄 Springer

accumulated errors impact the estimating accuracy greatly [4, 19, 21, 22]. GPS is another commonly used technique to solve this problem. However, GPS can't be used in some conditions such as indoor environments [25, 28]. Magazine [17] proposed The Non-Cooperative Feature Points for the motion estimation, which are implemented by extracting special satellite feature points that are Non-Cooperative for the implementation of navigation. The problem of this method is that it is limited in special environment. The graph-based methods [10] have been utilized for the robot localization and navigation, but this method is difficult to obtain the high accuracy. The technical, using Visual sensors to estimate camera's ego-motion, is called visual odometry. it is a promising technique for improving the capability of a mobile robot to estimate its motion in indoor environment [8, 11, 18, 24]. Extensive research has been done to make use of visual odometry in practice, several methods have been proposed based on monocular image sequences or stereo image sequences; however, there are still many difficulties in estimating the robots motion effectively because of the inaccurate feature matching and outliers [3, 14, 23]. One object of this paper is to develop a method integrating distinguished features of indoor environment and wireless sensor network to improve the effect of visual odometry. Considering that it is mainly applied to the indoor environment, the robot is assumed to travel in flat floor. The second problem needs to be solved is the robustness. Vision sensors are easy to be affected by many factors: the degree of image distortion correction, the tracking error of the feature points, and so on [1, 2, 26]. A series of filter algorithms are commonly used to improve the robustness of vision sensors, where particle filter has been a successful algorithm to solve this problem compared with other algorithms [6, 7, 12, 15]. However, A problem is difficult to be solved in particle filter algorithm that the particles would degenerate over time in terms of accuracy, which happens when the diversity of particles disappear over time. The main factors contributing to the disappearing of particles' diversity is in the resampling process: where the low weight particles would be thrown away and only can the particles with high weight survive. In recent years, many algorithms [5, 20] have been proposed to solve the problem of particle depletion. Several kinds of particle filter algorithms utilizing compensation techniques to deal with the depletion problem. Similarly, A new resampling technique is proposed to restrain the particle depletion using geometric relation between particles. Swarm intelligence is also integrated into the process of motion estimation, but the problem is that the unified framework is difficult to be constructed [9, 13, 16, 27]. These methods have shown better performance than original particle filter, However, they haven't taken particles' set as a whole and merely tried to maintain diversity. We propose an improved particle filter method to solve the particle depletion problem, which utilizes curve fitting to predict the trend of particle weights in resampling phase. The specific steps can be described as follows: Firstly, the discrete particles with different weights can be taken as a whole, and curve fitting is used to obtain the pattern of weight distribution; Then the resampling process can be carried out in this weight curve.

This paper is organized as follows: Section 2 described the main methods including the robot motion estimation and enhanced particle filter to improve estimation robustness; In Section 3, various experiments were implemented to verify the proposed method. Some difficult problems targeting at indoor environment were discussed in Section 4. Section 4 gives conclusions.

2 Methods

2.1 Motion estimation based on feature points of vision sensor

2.1.1 The design of robot motion estimation based on feature points

How to gain the accurate visual features are the most fundamental problem in robot localization and motion control. The artificial features are often used for the robust feature extraction, but they are difficult for the flexible robot application because they need Complex manual arrangement beforehand. The natural features are easy to be obtained, but the robustness of the natural features is difficult to be solved. So, the distinguished features combining the real environment like points and lines are the primary idea to describe the environments through vision sensors. Considering that a better result can be obtained if a priori knowledge of the environment is utilized in the feature extraction algorithm, a thorough comparison is implemented among various feature detection algorithms. In the real indoor environment, the straight lines in images are all parallel or vertical, and their parameter ranges can be determined based on the results in the previous frame; Therefore, the extraction of line features is of great significance to the environment understanding. In order to improve the feature robustness, the point features obtained by the cross of lines are appropriate for robot positioning. Finally, the standard Canny algorithm is selected for edge detection, and Hough transform is selected for line feature detection. The detailed process can be described as follow:

- The edge points are extracted in the whole indoor environment through canny algorithm when the robot is the first time to run.
- The lines are detected and the line parameter are obtained utilizing Hough Transform according to the results of edge contour.
- Based on the fact that the straight lines in images of indoor environment are all parallel or vertical, the parameter optimization is implemented through least square method.
- Finally, the point features through the cross of lines are chosen as the best features for robot positioning.

Hough transform has the advantage that it is not easily affected by illumination change, so the point features can be obtained in spite of the strong interference of light. In order to improve the robustness of features further, the homography matrix H is utilized to eliminate the outliers, which is calculated through the correspondent features from the last frame to the current frame. The complete process of homography matrix calculation is described as follows.

Two steps are applied to reject outliers. Firstly, an initial estimation of the homography matrix H is obtained through the RANSAC algorithm. Secondly, the unmatched features were eliminated according to the homography matrix H. The concrete homography matrix H calculation can be represented in following equations. Assuming that the correspondent feature points are $P_i(x_i,y_i), V_i(u_i,v_i)$ and P_i and V_i should satisfy Eq. 1:

$$k \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & m_9 \end{bmatrix} \tag{1}$$

Equation 2 can be obtained through Eliminating k from Eq. 1:

$$\mathbf{K} \cdot \mathbf{M} = m_9 \cdot \mathbf{U} \tag{2}$$

Where K, M and U are defined as follows:

$$\mathbf{K} = \begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i u_i & -y_i u_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -x_i v_i & -y_i v_i \\ & & & \cdots & & & & \\ & & & \cdots & & & & \end{bmatrix} \tag{3}$$

$$\mathbf{M} = \left(\mathbf{K}^T \mathbf{K}\right)^{-1} \mathbf{K}^T U \tag{4}$$

The relative motion will be very small when the two images are consecutive, therefore, we can obtain the motion speed according to the correspondences of the features in images. Based on the assumption of the planar ground, the world coordinate of points can be calculated according to the point coordinate projected in the image.

Point P is assumed as one point in the ceiling, and its coordinates are assumed as $p(u,v)$ on the image plane. The point coordinates can be converted to homogeneous coordinates, according to the linear camera model. The relations of points can be represented as Eq. 5:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{5}$$

Where M is a 3 * 4 projection matrix, which can be obtained through camera calibration. Because the point is on the ceiling, the Z axis coordinate of the point are set to 0. The equation can be rewritten as follows:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} m_{11} & m_{12} & m_{14} \\ m_{21} & m_{22} & m_{24} \\ m_{31} & m_{32} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathbf{m} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{6}$$

Among them, m is the remaining invertible 3×3 matrix after that the third column of matrix M is deleted. The upper formula shows that there is a one-to-one correspondence between point P and point p under the condition that the ceiling is flat, which means that the world coordinates can be derived from the Eq. 7 if the image point coordinates are given.

$$s \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathbf{m}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{7}$$

Therefore, as long as the coordinates of the corresponding feature points are obtained in two adjacent frames, it can be easy to estimate the robot motion, which can be represented as the motion of camera photocenter.

The process can be concluded as follows:

- To Capture the current frame of the video and rectify the image to remove the distortion.
- To Find the correspondent features in the current frame according to the features in the last frame
- To Compute the correspondent world coordinates of the feature points in the two frames
- To Infer the motion parameter according to distance the world coordinates of the correspondent feature points in the two frames.
- To Capture the next frame and repeat this process

2.1.2 The calculation of motion parameter

Based on the spatial analytic geometry theory, it needs at least 3 pairs of non-collinear matching points to determine the three-dimensional motion of the rigid body. Due to the special constraints of the indoor environment, the robot motion is in the two-dimensional plane. It only needs two points to determine the robot motion. In practical applications, in order to reduce the measurement error, the number of feature points selected is far greater than 2. A simple method for calculating the motion parameters is presented below.

$\{\mathbf{m}_i = (x_i, y_i)^T | i = 1, ..., n\}$ is assumed to be n (n >=2) feature points, where Z_i is zero and is deleted. R is assumed to be 2×2 rotation matrix; T is assumed to be 2×1 translation vector; $\{\mathbf{m}_i' = (x_i', y_i')^T | i = 1, ..., n\}$ is assumed to be $\{\mathbf{m}_i\}$ the matched points after motion,If the data does not contain noise, the Eq. 8 can be obtained:

$$\mathbf{m}_i' = \mathbf{R}\mathbf{m}_i + \mathbf{T}, i = 1, ...n \tag{8}$$

$\overline{\mathbf{m}'}$, $\overline{\mathbf{m}}$ are assumed as the gravity center of the two point sets respectively.

$$\overline{\mathbf{m}'} = \frac{\sum\limits_{i=1}^{n} \mathbf{m}_i'}{n} = \begin{bmatrix} \overline{x'} \\ \overline{y'} \end{bmatrix}, \quad \overline{\mathbf{m}} = \frac{\sum\limits_{i=1}^{n} \mathbf{m}_i}{n} = \begin{bmatrix} \overline{x} \\ \overline{y} \end{bmatrix} \tag{9}$$

Substitute $\overline{\mathbf{m}'}$, $\overline{\mathbf{m}}$ to Eq. 9:

$$\overline{\mathbf{m}'} = \mathbf{R}\overline{\mathbf{m}} + \mathbf{T} \tag{10}$$

Let Formula 9 subtract Formula 10,The following function can be referred:

$$\mathbf{m}_i' - \overline{\mathbf{m}'} = \mathbf{R}\left(\mathbf{m}_i - \overline{\mathbf{m}}\right) \quad i = 1, ...n \tag{11}$$

R is the rotation matrix and it can be solved by the least square method, and the translation vector can be obtained by the following equation.

$$\mathbf{T} = \overline{\mathbf{m}'} - \mathbf{R}\overline{\mathbf{m}} \tag{12}$$

This method is simple, but it has a serious problem: The matrix obtained is not a rotation matrix, because the constraints on the rotation matrix are not taken into account. The result is not satisfactory when the data contains noise.

Considering the constraint conditions of the rotation matrix R, we use the Euler angle to represent the rotation matrix.

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, 0 \le \theta < 2\pi \tag{13}$$

In order to calculate Euler angle θ, the minimum value of the following function should be calculated based on Eq. 13:

$$f(\theta) = \sum_{i=1}^{n} \left\| \mathbf{m}_i' - \overline{\mathbf{m}'} - \mathbf{R}\left(\mathbf{m}_i - \overline{\mathbf{m}}\right) \right\|^2 \tag{14}$$

It can be calculated to make the first derivative if the function is zero:

$$\tan\theta = \frac{\sum_{i=1}^{n}\left(\left(x_i - \overline{x}\right)\left(y_i' - \overline{y'}\right) - \left(y_i - \overline{y}\right)\left(x_i' - \overline{x'}\right)\right)}{\sum_{i=1}^{n}\left(\left(x_i - \overline{x}\right)\left(x_i' - \overline{x'}\right) + \left(y_i - \overline{y}\right)\left(y_i' - \overline{y'}\right)\right)} \tag{15}$$

Translation vector can be obtained based on Formula 13. After the motion parameter *(R, T)* is obtained, it can be used to get rid of the outliers. Considering the measurement error, Eq. 9 should be rewritten as:

$$m_i' = Rm_i + T + \eta_i \tag{16}$$

Where η_i represents the errors between the real position and the calculated position of feature points. The *n + 1* feature points are substituted to the upper formula in turn, and if the η_i modulus is larger than a certain value, the point can be considered as an outlier. After removing the outliers, the above formula can be reused to get more accurate motion parameters.

2.2 Particle filter for robust estimation

2.2.1 Particle filter with enhanced resampling

The general description of robot motion description is that we can estimate the state of feature point at a specific time using a set of camera sequence images, and then transform it into a posterior probability distribution function for target state. It is assumed that the change of the target state $p(x_k| x_{k-1})$ is Markoff process and the observed values $\{Z_1, Z_2 \ldots Z_k\}$ are independent of each other. Given the state transition prior probability $p(X_k|X_{k-1})$, where k denotes the time step. The observed likelihood function $p(X_k|Z_{1:k})$, the posterior probability distribution can be obtained based on Bayesian theory:

$$p(X_k|Z_{1:k}) = \frac{p(Z_k|X_k)\int p(X_k|X_{k-1})p(X_{k-1}|Z_{1:k-1})dX_{k-1}}{p(Z_k|X_k)\int p(X_k|X_{k-1})p(X_{k-1}|Z_{1:k-1})dX_{k-1}dX_k} \tag{17}$$

Because the calculation of state posterior probability contains complex and intractable integral operation, it is difficult to obtain its analytical solution. Monte Carlo technology is used to approximate the posterior state distribution with a set of weighted samples.

In recent years, in order to deal with the state estimation problem of nonlinear system and non-Gauss dynamic system, Particle Filter is also known as Bayesian Bootstrap Filter or

Monte Carlo Filter. The particle filter is the Bayesian framework under the theory of sequential importance sampling by Monte Carlo method, which is based on Monte Carlo stochastic simulation theory. The system posterior $p(X_k|Z_{1:k})$,where k denotes the time step, is represented by a set of weighted random sampling $\{x_k^i, w_k^i\}_{i=1}^N$ with probability distribution, and the new distribution produced by Bayesian iterative evolution from these random samples. It is impossible to get sampled particles directly from posteriori distribution in practice, so we often use the proposed distribution $q(X_k|X_{k-1}, Z_{1:k})$ to approximate the posterior distribution and to get a group of particles from it. From the moment *k-1* to the moment *k*, the *i-th* particle weight up can be updated based on formula 18:

$$w_k^i = w_{k-1}^i \frac{p(Z_k|X_k^i)p(X_k^i|X_{k-1}^i)}{q(X_k^i|X_{k-1}^i, Z_{1:k})} \tag{18}$$

The posteriori distribution $p(X_k|Z_{1:k})$ can be approximated as:

$$p(X_k|Z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(X_k - X_k^i) \tag{19}$$

The state function $f(X_k)$ can be approximated as:

$$E[f(X_k)] \approx \sum_{i=1}^N w_k^i f(X_k^i) \tag{20}$$

In the original particle filter algorithm, the resampling process is just to replicate the high weight particles. it is easy to meet the problem of particle depletion, and the particles would degenerate over time in terms of accuracy. we designed a new method in the resampling process of particle filter, where least square curve fitting is used to increase the particle diversity. The process of curve fitting used in resampling is described as follows:

$(x_i, f_i), (i = 1, 2, \cdots, m)$ is assumed as the given particles' state and their responding weights before resampling, The function needs to be found in set $\Phi = \text{Span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$ to satisfy the following Eq. 21.

$$S^*(x) = \sum_{k=0}^n a_k^* \varphi_k^*(x)(n < m) \tag{21}$$

The errors can be represented as Eq. 22:

$$\delta_i = S^*(x_i) - f_i(i = 1, 2, \cdots, m) \tag{22}$$

$S^*(x)$ can be obtained to satisfy Eq. 23:

$$\begin{aligned} \sum_{i=1}^m \delta_i^2 &= \sum_{i=1}^m \omega(x_i)\left[S^*(x_i) - f_i\right]^2 \\ &= \min_{S(x) \in \Phi} \sum_{i=1}^m \omega(x_i)[S(x_i) - f_i]^2 \end{aligned} \tag{23}$$

Where $\omega(x) \geq 0$ is the given weight function in set *[a, b]*. The above method of finding the approximation function $S^*(x)$ is the least square method of curve fitting. The function $S^*(x)$ that satisfies the relational expression 23 is the least square solution. For the given (x_i, f_i), the only function $S^*(x) = \sum_{k=0}^n a_k^* \varphi_k^*(x)$ exists in function set $\Phi = \text{Span}\{\varphi_0, \varphi_1 \cdots, \varphi_n\}$ to satisfy Eq. 23. The coefficients $a_0^*, a_1^*, \cdots, a_n^*$ of the least squares solution can be solved by the Eq. 24.

$$\sum_{k=0}^{n} \left(\varphi_k, \varphi_j \right) a_k = (f, \varphi) \quad (j = 0, 1, 2, \cdots, n) \tag{24}$$

In our designed solution, the algebraic polynomial fitting is selected as the basic functions, which is $\{\varphi_0, \varphi_1, \cdots, \varphi_n\} = \{1, x, x^2 \cdots, x^n\}$. So, the corresponding solution function 24 can be rewritten as:

$$\begin{bmatrix} \sum \omega_i & \sum \omega_i x_i & \cdots & \sum \omega_i x_i^n \\ \sum \omega_i x_i & \sum \omega_i x_i^2 & \cdots & \sum \omega_i x_i^{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum \omega_i x_i^n & \sum \omega_i x_i^{n+1} & \cdots & \sum \omega_i x_i^{2n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sum \omega_i f_i \\ \sum \omega_i x_i f_i \\ \vdots \\ \sum \omega_i x_i^n f_i \end{bmatrix} \tag{25}$$

where $\sum_{i=1}^{m}$ is simplified as $\sum$, as well as $\omega_i = \omega(x_i)$. $S^*(x) = \sum_{k=0}^{n} a_k^* x^k$ is the required polynomial.

The concrete description is as follows: firstly, based on the distinguished particles, curve fitting was utilized to fit the trend of the particles with different weights. Secondly, the fixed number of particles was resampled based on the fitted curve. The illustrative diagram is shown in Fig. 1. Our method has two advantages: the first one is that the curve can reflect the trend of particle weight; the second one is that new particles can be easily resampled from the fitted curve, solving the problem of particle depletion. The whole process can be concluded as follows:

- Particles are used to simulate robot real state, which are assumed to be distributed uniformly in state space
- Each particle gets a predicted state according to the state transfer equation
- The predicted states of particles are evaluated and updated according to the observation
- The least square curve fitting is carried out according to the state and weight of the particles. The new resampling particles gets the weight from the fitted curve.
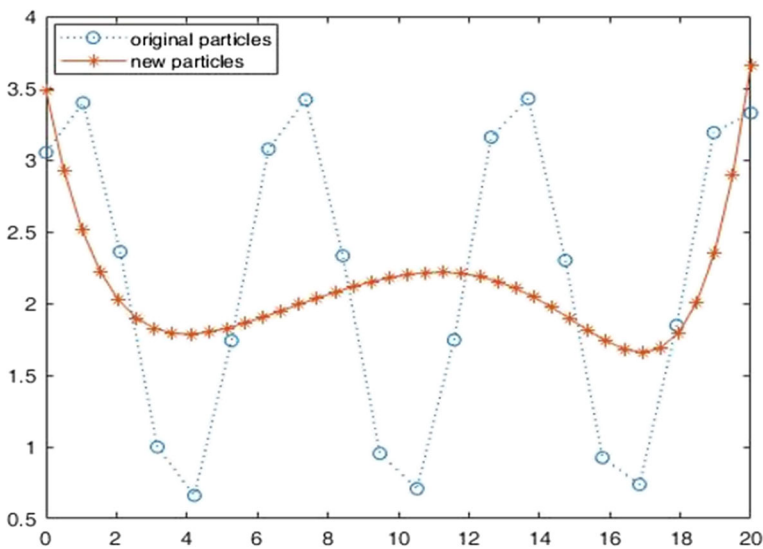

Fig. 1 The Illustrative diagram for particle resampling: the blue circle points denote the original particles and the red points denotes the new resampling particles

2.2.2 Observation model of feature points

Many factors affect the visual motion estimation errors, such as the light changes, running distance. For the robust motion estimation, The errors modeling methods are indispensable, which are usually set up vision imaging model according to computer vision theory, but it is difficult to adapt to the specific circumstances of indoor environment.

The Robot observation model is shown as Fig. 2, where the pentagram denotes the real robot position, while the observation results have some errors shown as asteroids. Considering the efficiency, A direct method is used to set up the vision errors model through the analysis of the experimental data. Our vision error model is built as Eq. 26:

$$z_i = x_i + v_i \tag{26}$$

The observation noise v_i is also assumed to be Gaussian noise with the mean value of zero, and the covariance matrix R_i is a diagonal, where the diagonal elements are as follows:

$$\begin{aligned} R_{11} &= \sigma_{rx}, \\ R_{22} &= \sigma_{ry}, \\ R_{33} &= \sigma_{\theta} \end{aligned} \tag{27}$$

According to the measurement results, the variance of the position distribution σ_x were obtained using curve fitting, where r is the distance from the current point to the feature point. Curve fitting is used because the variance will turn very great in the blind area without feature points. The change of angle variance σ_{θ} is not obvious with the change of distance.
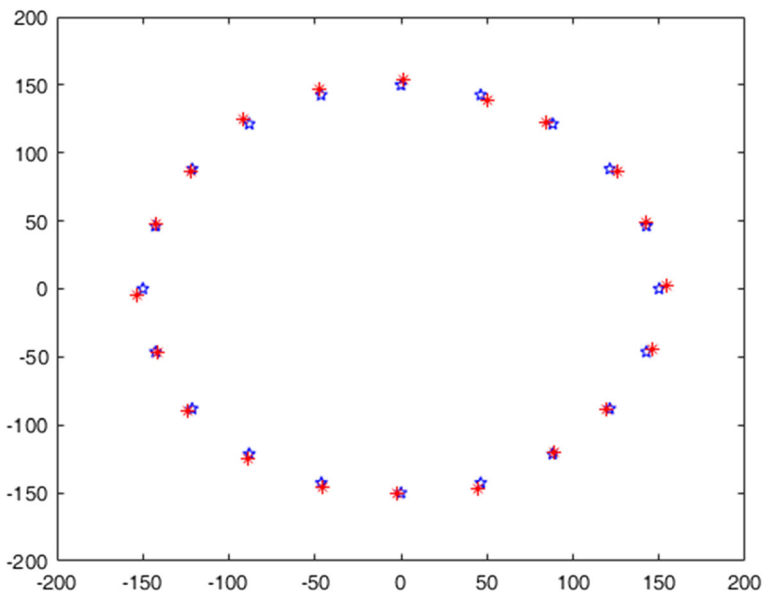

Fig. 2 The illustrative graph for robot observation error: the pentagram denotes the real robot position and the asteroid denotes the robot observation position. The unit is cm
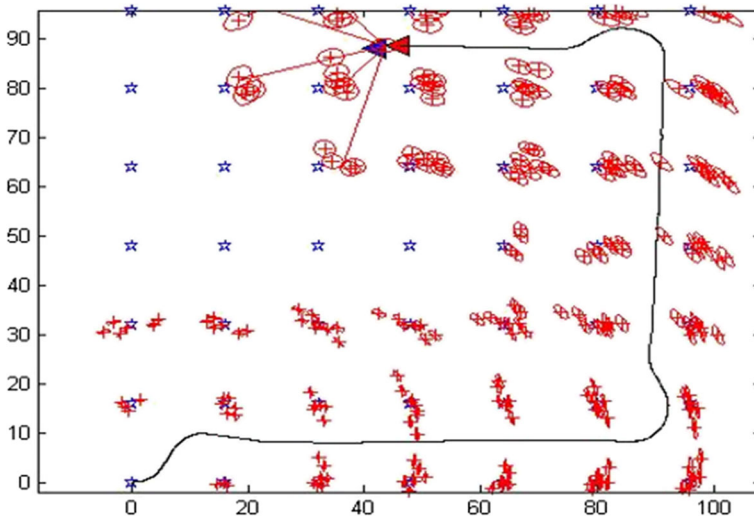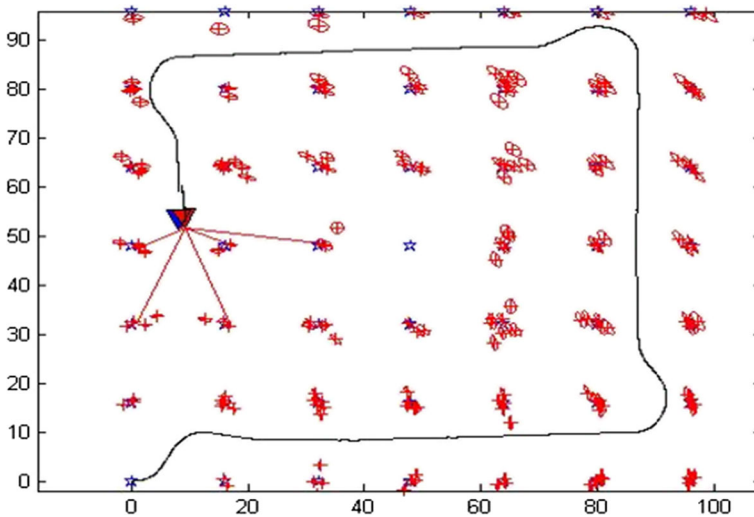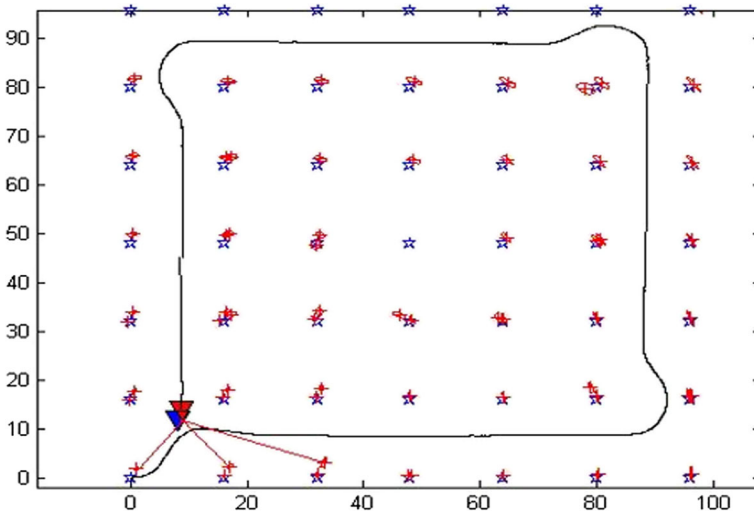
Fig. 3 Result of UKF algorithm: The axis X and axis Y denote the room length and room width respectively, and the unit is 5 cm

3 Results

The Institute of automation of Chinese Academy of Sciences has independently developed a versatile autonomous mobile robot platform called AIM based on its existing technology. The various theories and algorithms described in this paper are tested on the platform. The vision sensor is placed on the top of the mobile robot.

In order to obtain the observation model of vision sensors, A set of measurement values and errors are obtained through actual experiments, where the real position of the feature point is obtained by manual measurement. The measured points are distributed equally in three sets of

Fig. 4 Result of original particle filter experiment: The axis X and axis Y denote the room length and room width respectively, and the unit is 5 cm

Fig. 5 Experiment result of proposed algorithm called PFCF: The axis X and axis Y denote the room length and room width respectively, and the unit is 5 cm

concentric circles, and the feature points to be observed are located at the center of the circle. The measurement error varies with different distances to feature points. Farther away from the feature points, the more measurement errors, which is caused by image distortion. Though the camera calibration and image correction have been implemented, the absolutely accurate camera calibration is difficult to achieve. Based on this phenomenon, our observation model is set up considering more on the actual environment. The experiment environment is an indoor room with area 5 m by 5 m, which is shown in Fig. 3. The pentagram represents the feature points using the cross of vertical lines extracted by Hough transform. The center of red eclipses represents the feature points detected, and the area of a certain eclipse represents the error variance corresponding to a certain feature point. The more the area, the more the error variance. The robot path is scheduled as follows: The robot, which is represented by the purple triangle, starts from the corner and runs around the rectangular wall. When the robot detects a feature point, it estimates the feature position and the corresponding error variance. Simultaneously, it estimates its own position according to the detected feature points. The motion speed will be determined based on its position in different time. Because the errors in the estimation of the feature points, the estimation of robot's own position would have errors too. The estimation position of the robot is represented by the red triangle.

The UKF [13] is selected for the comparison of experiment results, which is the abbreviation of the Unscented Kalman Filter. In order to explain the results in more detail, the original particle filter [16] is also selected. Because our proposed algorithm combines the original particle filter and cure fitting, it is represented by PFCF (Particle Filter with Curve Fitting). UKF and original

Table 1 Standard deviation of particle position

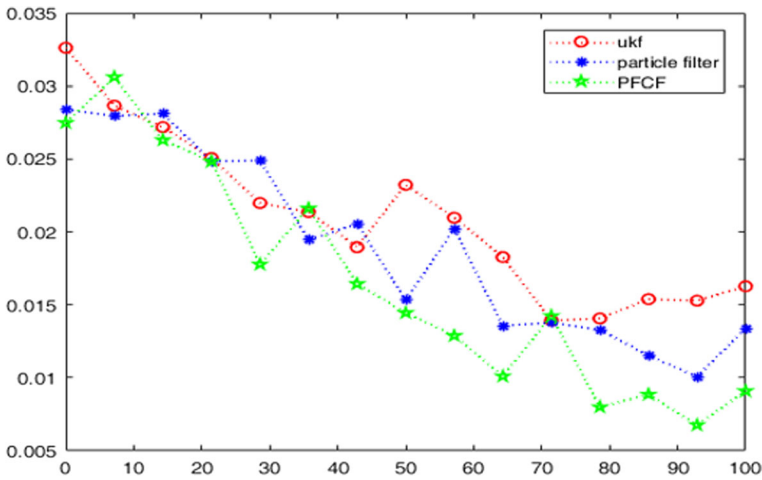| Type	Position(m)	Feature(m)
UKF	0.083	0.079
Particle filter	0.074	0.068
PFCF	0.060	0.057

Fig. 6 The average values of feature error with different numbers of particles: The axis X denotes the numbers of particles, and the axis Y denotes average values of feature error (m)

particle filter are shown in Figs. 3 and 4 respectively. Figure 5 shows the result of PFCF. Apparently, the error of landmark estimation becomes smaller in our proposed algorithm than that in original particle filter and UKF. The standard deviations of estimation errors for different algorithms are summarized in Table 1. it is worth noting that each value in this table averages the whole results for different numbers of particles. The standard deviations of original particle filter are smaller than UKF, and the standard deviations of PFCF are smaller than original particle filter.

Different numbers of particles have important impacts on the estimation of feature error and position error for different algorithms. For better comparison, we analyze the results of different algorithms from different angles. The average values of feature error for different algorithms are shown in Fig. 6. The average feature error turns smaller with the number of particles increases. Among the Three algorithms, the worst algorithm is UKF, which has the
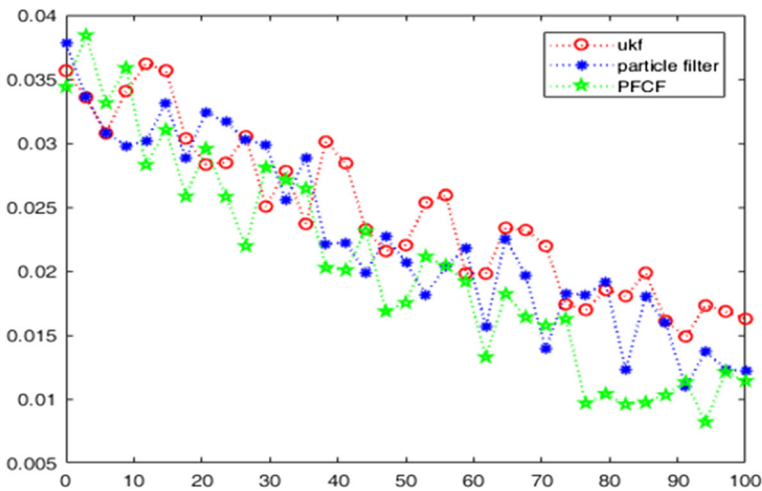

Fig. 7 The average values of position error with different numbers of particles: The axis X denotes the numbers of particles, and the axis Y denotes average values of position error (m)
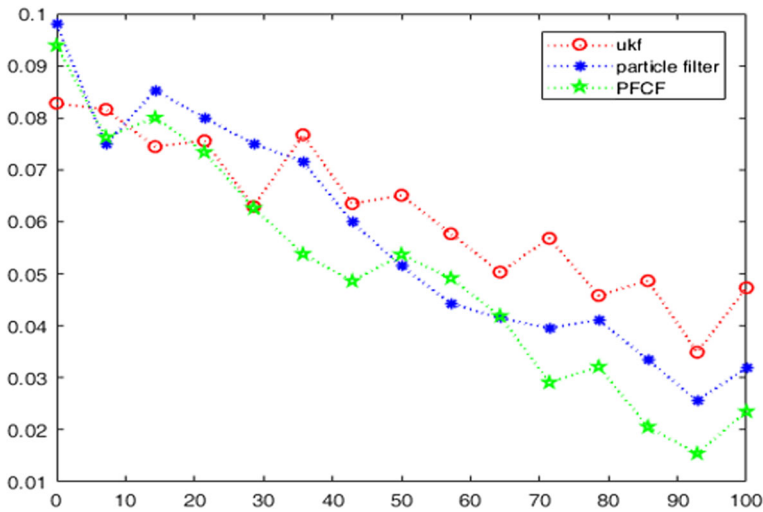
Fig. 8 The maximum values of feature error with different numbers of particles: The axis X denotes the numbers of particles, and the axis Y denotes maximum values of feature error (m)

largest feature estimation errors; while the best algorithm is the proposed PFCF, which has the smallest feature estimation errors. The comparison is carried out in the same particle number. The average position error values for different algorithms and different numbers of particles are shown in Fig. 7. It is shown that the average position errors of the three algorithms become reduced as the number of particles increases. The worst algorithm is UKF, which has the largest average position error, and the best algorithm is still the proposed PFCF, which has the smallest position estimation error for the same particle number.

Figure 8 shows the impact of different numbers of particles in maximum estimation of feature position, and Fig. 9 shows that of position. In these conditions, PFCF has the best results for each same number of particles, and particle filter is better than the UKF. The experiment
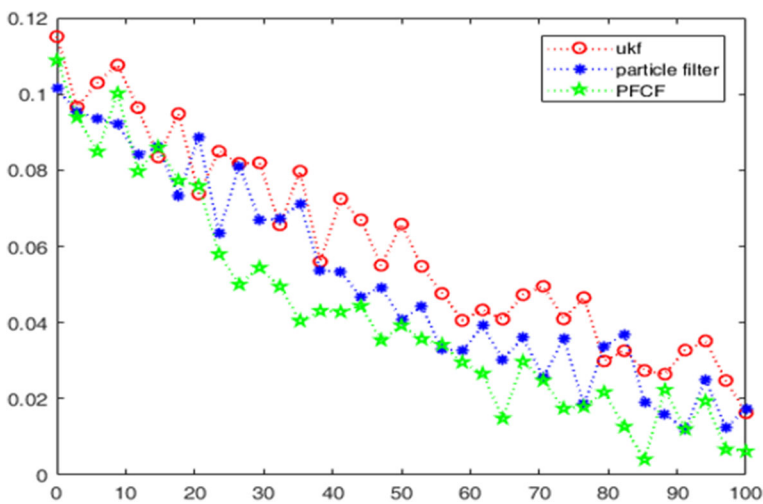

Fig. 9 The maximum values of position error with different numbers of particles: The axis X denotes the numbers of particles, and the axis Y denotes maximum values of position error (m)

results show that conventional particle filter degenerates over time in terms of accuracy due to the particle depletion in resampling process. The curve fit method is utilized in PFCF, which can represent the trend of the particle weights, to solve the particle depletion problem. The results show that it can improve the performance of particle filter, contributing to the better estimation results. Besides, there are many factors that will affect the algorithm, such as the degree of image distortion correction, the jitter of the camera in the robot's running process, the tracking error of the feature point, and the pitching of the robot during the emergency stop and acceleration. Some of these factors have a great impact on the results of the algorithm, such as the jitter of the camera during the operation and the pitching of the robot. The future work will focus on these problems, and the depth information acquisition and matching for the real indoor environment will be taken into account to get more robust estimation results.

4 Conclusions

Future internet of things relies on the various robot services, while robot localization and motion estimation are key factors to implement different robot applications. The method utilizing traditional inertial sensor is easy to be impacted by the wheel slipping, and the accumulated error is a serious problem to be resolved when the robot moves a long time. Vision sensors with huge amount information have been one of the most important sensors in the various applications of the mobile robot. The indoor environment is semi-structured environment, So the algorithm combining the characteristics of indoor environment can bring a lot of convenience, which can also help to improve the accuracy, real-time and reliability of robot localization and motion estimation. An effective robot motion estimation method is proposed assuming that the mobile robot runs on the flat ground. Because the visual features are easily affected by the changes of light, an enhanced particle filter method integrated with least square curve fitting is designed to improve the robustness. The experiments on the wheeled robot verified that the algorithm can be carried out reliably for robot motion and positioning estimation in the indoor environment. The results also show that our proposed enhanced particle filter can perform better than the original particle filter. Future work will integrate the depth information with the real indoor environment to implement the more robust motion estimation.

Author's contributions Rong Jiang, Xiaoming Dong designed the experiments and wrote the paper; Rong Jiang, Xiaoming Dong performed the experiments; Liefu Ai analyzed the data; All authors have read and approved the final manuscript.

Compliance with ethical standards

Competing interests The authors declare that they have no competing interests.

References

1. Alon J, Sclaroff S (2000) Recursive estimation of motion and planar structure. In: IEEE computer vision and pattern recognition, pages II:550–556
2. Ascani A, Frontoni E, Mancini A 2008 Feature group matching for appearance-based localization. In: IROS
3. Azarbayejani A, Pentland AP (1995) Recursive estimation of motion, structure, and focal length, IEEE transactions on pattern analysis and machine intelligence. 17(6):562–575
4. Bortz JE (2007) A new mathematical formulation for strapdown inertial navigation. IEEE Trans Aerosp Electron Syst AES-7(1):61–66
5. Bronzwaer S, Cars O, Buchholz U et al (2002) A European study on the relationship of antimicrobial use and antimicrobial resistance[J]. Emerg Infect Dis 8(3):278–282
6. Campbell J, Sukthankar R, Nourbakhsh I (2004) Techniques for evaluating optical flow in extreme terrain. Proceedings of IROS
7. Carlevaris-Bianco N, Kaess M, Eustice RM (2017) Generic node removal for factor-graph SLAM[J]. IEEE Trans Robot 30(6):1371–1385
8. Cumani A, Guiducci A (2009) ,Comparison of feature detectors for rover navigation. MMCTEE pp.126-131.
9. Davison AJ, Reid ID, Molton ND et al (2007) MonoSLAM: real-time single camera SLAM[J]. IEEE Trans Pattern Anal Mach Intell 29(6):1052–1067
10. Eckenhoff K, Paull L, Huang G (2016) Decoupled, consistent node removal and edge sparsification for graph-based SLAM[C]. IEEE international conference on intelligent robots and systems pp 3275–3282
11. Helmick DM, Chang Y, Roumeliotis SI, Clouse D, Matthies L (2004) Path following using visual odometry for a mars rover in high-slip environments. In: Proc. 2004 IEEE Aerospace Conference, Big Sky, MT,Mar. 6–13, 2004
12. Indelman V (2017) No correlations involved: decision making under uncertainty in a conservative sparse information space[J]. IEEE Robot Autom Lett 1(1):407–414
13. Kandepu R, Foss B, Imsland L (2008) Applying the unscented Kalman filter for nonlinear state estimation [J]. J Process Control 18(7):753–768
14. Lowe DG (2004) Distinctive image features from scale invariant keypoints. Int J Comput Vis 60(2):91–110
15. Magnabosco M, Breckon TP (2013) Cross-spectral visual simultaneous localization and mapping (SLAM) with sensor handover. Robot Auton Syst 63(2):195–208
16. Montemerlo M, Thrun S, Koller D et al (2002) FastSLAM: a factored solution to the simultaneous localization and mapping problem[C], AAAI pp 593–598
17. Ning M, Zhang S, Wang S (2018) A non-cooperative satellite feature point selection method for vision-based navigation system. Sensors 18:854
18. Olson C F, Matthies L H, Schoppers M, et al. (2003). Rover Navigation using Stereo Ego-motion[J]. Rob Auton Syst 43(4):215–229.
19. Potkonjak M, Feng J, Girod L (2006) Location discovery using data drive statistical error modeling[J]. IEEE FOCOM 26(8):43–47
20. Sajeeb R, Manohar CS, Roy D (2009) Rao-Blackwellization with substructuring for state and parameter estimations of a class of nonlinear dynamical systems. IJEUU 1:1–2
21. Sirtkaya S, Seymen B, Alatan AA (2013) Loosely coupled Kalman filtering for fusion of visual odometry and inertial navigation. In: Proceedings of the 2013 16th international conference on information fusion (FUSION), Istanbul, Turkey, 9–12 July 2013, pp 219–226
22. Ssu K-F, Ou C-H, Jiau HC (2005) Localization with mobile anchor points in wireless sensor networks[J]. IEEE Trans Veh Technol 54(3):1187–1197
23. Torr PHS, Fitzgibbon A, Zisserman A (1999) The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. Int J Comput Vis 32(1):27–45
24. Wang H, Yuan K, Zou W, Zhou Q (2005) Visual odometry based on locally planar ground assumption, IEEE International Conference on Information Acquisition(ICIA2005)
25. Wang C, Wang T, Liang J, Chen Y, Wu Y (2012) Monocular vision and IMU based navigation for a small unmanned helicopter. In: Proceedings of the 2012 7th IEEE conference on Industrial Electronics and Applications (ICIEA), Singapore, 18–20 July 2012, pp 1694–1699

26. Wu J, Li J, Chen W (2017) Practical adaptive fuzzy tracking control for a class of perturbed nonlinear systems with backlash nonlinearity. Inf Sci 420:517–531
27. Wu J, Wu Z-G, Li J, Wang G, Zhao H, Chen W (2018) Practical adaptive fuzzy control of nonlinear pure-feedback systems with quantized nonlinearity input. IEEE Trans Syst Man Cybern Syst Hum. https://doi.org/10.1109/TSMC.2018.2800783
28. Xu D, Han L, Tan M, Li YF (2009) Ceiling-based Visual Positioning for an Indoor Mobile Robot with Monocular Vision. IEEE Trans Ind Electron 56(5):1617–1628

Xiaoming Dong received his Ph.D. degree from Automation of Institution, Chinese Academy of Sciences in 2010. He is currently an associate professor at the school of computer and information, Anqing Normal University and the key laboratory of intelligent perception and computing of Anhui Province. His research interests lie in robot navigation, wireless sensor networks, pattern recognition, image processing and Internet of things etc. (E-mail: dongxiaoming@aqnu.edu.cn)

Liefu Ai received the Ph. D. degree from Huazhong University of Science and Technology in 2014. He is currently an associate professor at the school of computer and information, Anqing Normal University and the key laboratory of intelligent perception and computing of Anhui Province. His research interests lie in machine learning, pattern recognition etc. (E-mail: ailiefu@qq.com)
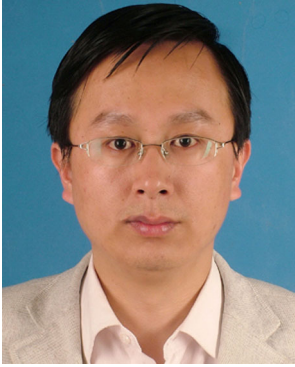
Rong Jiang* received his Ph.D. degree in system analysis and integration from the school of software at Yunnan University. He is currently a professor and Ph. D. supervisor at the school of information, Yunnan University of Finance and Economics, China. He has published more than 30 papers and ten books and has gotten more than 40 prizes in recent years. His main research interests include cloud computing, big data, software engineering, information management, pattern recognition etc. (E-mail: jiang_rong@aliyun.com)