Binocular Vision Pose Estimation Based on PSOPF

Yu Fei, Qiang Fu, Ying Zhuang, Xiang Liu, Gonglei Liao and Qichun Zhao,
Sichuan Provincial Machinery Research & Design Institute, Chengdu 610063, China
Email: 1726691225@qq.com

Abstract. In order to realize robot autonomous localization and navigation, binocular vision
localization based on particle swarm optimized particle filter (PSOPF) is put forward according to the
characteristics of nonlinear and non-Gaussian distribution complex system. The localization of 6-DOF
robot only depends on binocular vision in the method. At first, road signs are obtained by SIFT feature
matching points of binocular vision. The second, initial pose estimation is obtained by four elements.
Finally, robot pose is estimated accurately by PSOPF and the algorithm overcomes the shortcoming
of particle filter (PF) and improve estimation accuracy. Experiment results show that this algorithm
has high computing accuracy and robustness.

1. Introduction
Visual sensors are more and more used in the localization of robot in complex unpredictable environment
with the development of visual technology [1-4]. There are convergency problem in iterative algorithm and
degeneracy problem in particle filter. PSOPF is put forward in order to solve these problems in this paper.
Particle swarm optimization (PSO) is put forward by Kennedy and Eberhart [5] in 1995, which is an
intelligence optimization algorithm by simulating the animal swarm behavior. PF has many similarities with
PSO. At first, PSO searches the optimal value through continuous updating speed and position of particles,
and PF approximates real posterior probability distribution through continuous updating position and weight
value of particles. Secondly, the particle of largest fitness value is the optimum value in the search space in
PSO, and the particle of maximum weight is the most like likely state of the system in PF. At last, PSO algorithm
and PF algorithm have their own movement mechanisms. Particle updates its speed and position through the
pursuit of individual optimal value and the global optimal value in PSO, and particle updates own pose by
motion model and updates own weight value by observation model in PF.

2. Binocular Vision Location Running Processes
As shown in figure 1, robot location is realized by continuous image frame information of the surrounding
environment of path. Camera calibration and image distortion should be satisfactorily solved to avoid the
image information error accumulation. Road signs are obtained by SIFT feature matching points of binocular
vision. The initial pose estimation of robot is obtained by the method of four elements and robot pose is
estimated accurately by PSOPF.

Figure 1. Binocular vision location running processes
3. Initial Pose Estimation Based on Four Elements
The road signs (ID1, ID2, ID3, ID4...) obtained by binocular vision at the different moments are shown in figure 2. The three dimensional coordinates are \( \{X_1t, X_2t, X_3t, X_4t, \ldots\} \), \( \{X_1t+1, X_2t+1, X_3t+1, X_4t+1, \ldots\} \). The pose increment of robot between the two different moments is easily evaluated by unit quaternion.

![Figure 2. Scheme of the binocular vision pose estimation](image)

The feature points of images based on SIFT are taken as road signs [6]. The robot pose estimation is realized by four elements with the matching road signs of the two different moments [7]. The pose increment of robot is given as

\[
\begin{align*}
\mu_q &= (\Delta x, \Delta y, \Delta z, \Delta \alpha, \Delta \beta, \Delta \gamma) \\
(1)
\end{align*}
\]

where \( (\Delta x, \Delta y, \Delta z) \) are the translation increments, and \( \Delta \alpha, \Delta \beta, \Delta \gamma \) are rotation increments.

The calculation of first element is as follows:

\[
\begin{align*}
&c_i = \frac{1}{N} \sum_{i=1}^{N} X'_{i} \\
c_{i+1} = \frac{1}{N} \sum_{i=1}^{N} X'_{i+1} \\
X'_i = (X'_i, Y'_i, Z'_i) = X'_i - c_i \\
X'_{i+1} = (X'_{i+1}, Y'_{i+1}, Z'_{i+1}) = X'_{i+1} - c_{i+1}
\end{align*}
\]

(2)

The calculation of second element is as follows:

\[
\begin{align*}
P_{XX'} &= \bar{X}_i' X'_{i+1} \\
P_{XY'} &= \bar{Y}_i' X'_{i+1} \\
P_{XZ'} &= \bar{Z}_i' X'_{i+1} \\
P_{YY'} &= \bar{Y}_i' Y'_{i+1} \\
P_{YZ'} &= \bar{Z}_i' Y'_{i+1} \\
P_{ZZ'} &= \bar{Z}_i' Z'_{i+1} \\
P_{XX} &= \bar{X}_i X_{i+1} \\
P_{XY} &= \bar{Y}_i Y_{i+1} \\
P_{XZ} &= \bar{Z}_i Z_{i+1} \\
P_{YY} &= \bar{Y}_i Y_{i+1} \\
P_{YZ} &= \bar{Z}_i Z_{i+1} \\
P_{ZZ} &= \bar{Z}_i Z_{i+1}
\end{align*}
\]

(3)

The calculation of third element is as follows:
The forth element is as follows:

\[
S_{xx} = \frac{1}{n} \sum_i P'_{XX}, \quad S_{xy} = \frac{1}{n} \sum_i P'_{XY}, \\
S_{yx} = \frac{1}{n} \sum_i P'_{YX}, \quad S_{yy} = \frac{1}{n} \sum_i P'_{YY}, \\
S_{xz} = \frac{1}{n} \sum_i P'_{XZ}, \quad S_{yz} = \frac{1}{n} \sum_i P'_{YZ}, \\
S_{zx} = \frac{1}{n} \sum_i P'_{ZX}, \quad S_{zy} = \frac{1}{n} \sum_i P'_{ZY}.
\]

(4)

The forth element is as follows:

\[
N = \begin{pmatrix}
N_{11} & N_{12} & N_{13} & N_{14} \\
N_{21} & N_{22} & N_{23} & N_{24} \\
N_{31} & N_{32} & N_{33} & N_{34} \\
N_{41} & N_{42} & N_{43} & N_{44}
\end{pmatrix}
\]

(5)

Then the relationship between rotation matrix and eigenvalues is:

\[
R(\alpha) = \begin{pmatrix}
n_0^2 + n_1^2 - n_2^2 - n_3^2 & 2(n_0 n_1 - n_2 n_3) & 2(n_0 n_2 + n_1 n_3) \\
2(n_0 n_1 + n_2 n_3) & n_0^2 - n_1^2 + n_2^2 - n_3^2 & 2(n_0 n_3 - n_1 n_2) \\
2(n_0 n_2 - n_1 n_3) & 2(n_0 n_3 + n_1 n_2) & n_0^2 + n_1^2 - n_2^2 + n_3^2
\end{pmatrix}
\]

The pose increment of robot in translation is:

\[
t = c_{t_i} - R c_{t_{i+1}}
\]

4. Accurate Pose Estimation by PSOPF

At first, sample pose particles are optimized by PSO. The second, the weights of optimized particles are calculated as the samples of PF. Each particle shows a possibility of robot pose in PF.

**Step 1**: particle sampling

The initial pose by four elements is sampled by random sampling method. The particle swarm of size N is generated in space: \( H = \{h_1, h_2, \ldots, h_N\} \).

Where \( h \) is six dimensional vector of the particle in space:

\[
h = (\alpha, \beta, \gamma, t_x, t_y, t_z)
\]

**Step 2**: the fitness function calculation

\[
E(h) = \frac{1}{\sqrt{\frac{1}{n} \sum_{i=1}^{n} || P_i - m_i ||^2}}
\]

(6)

**Step 3**: The prediction value of road signs

The prediction value of the road sign is given as

\[
\overline{m_i} = R_i X_i + T_i
\]

(7)

where \( R_i \) is relatively rotation matrix of each pose particle, and \( T_i \) is translation vector.

**Step 4**: The observation value of road signs

The observation value of road sign is calculated by

\[
X = (c - c_0) \frac{b}{d}; \quad Y = (r - r_0) \frac{b}{d}; \quad Z = f \frac{b}{d}
\]

**Step 5**: The particle swarm update
In order to effectively control particle migration speed and let the algorithm have fine search ability, inertia factor is introduced in the particle swarm update by borrowing the idea of simulated annealing. The updating rules of position and speed are as fellows.

\[
\begin{align*}
\vec{v}_{id}^{k+1} &= \omega \vec{v}_{id}^{k} + c_1 r_1 (\vec{P}_d^k - \vec{h}_{id}^k) + c_2 r_2 (\vec{P}_d^k - \vec{h}_{id}^k) \\
\vec{h}_{id}^{k+1} &= \vec{h}_{id}^k + \vec{v}_{id}^{k+1}
\end{align*}
\] (8)

In expression (8), \(\vec{v}_{id}^{k+1}\) is migration speed of the dimension \(d\) of the step \(k\) of the particle \(i\), \(c_1\) is cognition factor and \(c_2\) is society factor, \(r_1\) and \(r_2\) are random numbers of between 0 and 1. \(\vec{P}_d^k\) is individual extremum of dimension \(d\) of the particle \(i\). \(\vec{P}_d^k\) is the global extremum of the dimension \(d\), \(\omega\) is inertia factor. Generally, \(\omega = b - k (b - a) / K\), \(b = 0.9\), \(a = 0.4\), and \(K\) is the largest number of migration.

**Step 6:** Particle swarm update end

PSO is terminated when the number of migration reaches the set value \(K\) or particle fitness reaches the set value \(\zeta\).

**Step 7:** Calculation of the road sign prediction value of pose particle after PSO by expression (7).

**Step 8:** Particle weight calculation and normalization processing

The probability of each particle is given as:

\[
p(P_i | h_t) = \prod_j p(P_{ij} | h_t)
\] (9)

where \(p(P_i | h_t)\) is the influence of each road sign \(P_i\) for each particle \(h_t\). Assuming \(p(P_i | h_t)\) obeys Gaussian distribution. \(p(P_{ij} | h_t)\) is given as:

\[
p(P_{ij} | h_t) = N(\vec{P}_{ij}, \vec{m}_t, \sum_j + \sum_i)
\]

\[
= (2\pi)^{\frac{1}{2}} \sum_j + \sum_i \frac{1}{2} \exp \left(-\frac{1}{2} (\vec{P}_{ij} - \vec{m}_t)^T (\sum_j + \sum_i)^{-1} (\vec{P}_{ij} - \vec{m}_t) \right)
\] (10)

The calculation of particle weight is given as:

\[
\omega_{ij}^{[k]} = \omega_{ij}^{[k-1]} p(P_i | h_{ij}^{[k-1]})
\] (11)

where \(\omega_{ij}^{[k]}\) and \(\omega_{ij}^{[k-1]}\) are the weight pose particle at time \(t\) and time \(t-1\). The normalized weight is given as

\[
\hat{\omega}_{ih}^{[k]} = \frac{\omega_{ij}^{[k]}}{\sum_{i=1}^{N} \omega_{ij}^{[k]}}
\] (12)

**Step 9:** particle self-adaption resampling

**Step 10:** The precise pose estimation

The precise pose estimation is given as:

\[
\hat{h}_t = \sum_{k=1}^{N} \hat{\omega}_{ih}^{[k]} h_{ij}^{[k]}
\] (13)

And the covariance is given as:

\[
\text{cov}_t = \frac{1}{N} \sum_{k=1}^{N} (\hat{h}_t - h_{ij}^{[k]}) (\hat{h}_t - h_{ij}^{[k]})^T
\] (14)

5. Experimental Studies

The experiment platform is shown in figure 3.
The line speed of the robot motion is 0.1m/s, and the maximum angular velocity is 0.3rad/s. The location accuracy of PSOPF is compared with orthogonal iterative algorithm and PF algorithm. The location errors of X axis and Y axis are shown in figure 4 and figure 5. Figure 6 shows rotation error of angle. Detailed values are shown in table 1.
6. Conclusion
In this paper robot autonomous localization is realized by binocular vision systems only, which has low cost and high efficiency. In order to improve the location accuracy, PSOPF is used in this paper, which avoids the divergence and local-minima problems in PF. The system can not only realize localization but also avoid obstacles. The system is very suitable for foot robot and the wet and slippery environment of road. Experiment results show that this system has a high accuracy and robustness.

7. Acknowledgment
We would like to appreciate the support by 2016 Intelligent Agricultural Machinery Equipment Project of Ministry of Science and Technology of China(2016YFD0700400) and Supported by Sichuan Economic and Information Commission([2018]53) .

8. References
[1] XU Y.X., JIANG Y.L., et al. (2011). Stereo visual localization based on generalized orthogonal iterative algorithm. ACTA PHOTONICA SINICA, 40(8), 1225-28.
[2] CHAO F.P., WANG R.B. (2011). Stereo vision based ego-motion estimation algorithm for lunar rover. Journal of Jilin University (Engineering and Technology Edition), 41(6), 1593-96.
[3] Moreno F.A., Blanco J.L., Gonzalez J. (2009). Binocular vision specific models for particle filter-based SLAM. Robotics and Autonomous Systems, 57,955-970.
[4] Pedro N., Ricardo V.M., Antonio B. (2011). Visual Odometry Based on Structural Matching of Local Invariant Features Using Stereo Camera Sensor. Sensors, 11, 7262-84.
[5] Kennedy J., Eberhart R. (1995). Particle swarm optimization. Proc of the IEEE Int Conf on Neural Networks, Piscataway: IEEE Service Center, 1941-48.
[6] Yang X.F., Huang Y.M. (2012). A Method for Improving Matching Efficiency of SIFT Features. China Mechanical Engineering, 23(11), 1297-99.
[7] ZHANG R.H., JIA H.G., CHEN T., et al. (2008). Attitude solution for strapdown inertial navigation system based on quaternion algorithm. Optics and Precision Engineering, 10, 1963-70.

Table 1. Algorithm comparison

| Algorithm  | X RMSE (m) | Y RMSE (m) | Angle RMSE (rad) |
|------------|------------|------------|-----------------|
| Orthogonal Iterative  | 0.336      | 0.342      | 0.0215          |
| PF         | 0.291      | 0.305      | 0.0181          |
| PSOPF      | 0.224      | 0.238      | 0.0172          |