Robust Binocular Pose Estimation Based on Pigeon-Inspired Optimization

Lu Gan, Haibin Duan, *Senior Member, IEEE*
State Key Lab of Virtual Reality Technology and Systems
School of Automation Science and Electrical Engineering
Beihang University (BUAA)
Beijing, China
hbduan@buaa.edu.cn

***Abstract*—In this paper, an accurate and robust pose estimation algorithm for binocular camera systems is proposed based on pigeon-inspired optimization (PIO), which can be easily generalized to estimation for multiple camera systems. In this method, the information of both cameras is completely used, and the poses of them can be determined accurately at the same time. Pigeon-inspired optimization is a new and efficient evolutionary algorithm, which is employed to select the optimal rotation axis in our method. The robustness and high accuracy of the proposed method has been demonstrated by three cases of experiments over some state-of-the-art pose estimation methods.**

Keywords*—pose estimation; binocular camera systems; multiple camera systems (MCS); pigeon-inspired optimization (PIO)

I. Introduction

Pose estimation of a calibrated camera is a fundamental problem and one of the most active research topics in computer vision [1]. It has numerous applications in augmented reality [2], [3], photogrammetry [4], and robotics [5], [6]. Pose estimation aims at retrieving the rigid transformation between the camera and the reference frame, given some 3D reference points and their corresponding projections on the image plane. Therefore, pose estimation is known as "perspective-*n*-point" (P*n*P) problem. To tackle this problem, various techniques have been well established and thoroughly studied. Many techniques are iterative and use a nonlinear optimization scheme [7]-[9], while others try to solve this problem by non-iterative methods to relieve the computational burden [10]-[12]. Generally, the iterative methods are more stable and accurate but more time-consuming than the non-iterative ones.

Recently, a robust non-iterative solution of P*n*P (RP*n*P) is proposed with the lowest computational complexity $\Theta(n)$ [10]. This solution is obtained by establishing a seventh order polynomial as a cost function and solving it directly. This pose estimator achieves the best performances among the existing non-iterative algorithms and as accurate results as the best iterative solutions. In addition, it is efficient and robust to work well in different configurations of reference points [10].

Previous investigations have focused on pose estimation from monocular view image sequences. In recent years, with the development and wide applications of multiple camera systems (MCS), interests have grown in pose estimation in the multi-view cases [13]-[16]. MCS can provide wider filed-of-view than a single camera, and have been commonly used for scene reconstruction as they can recover stereo information easily. Pose estimation for MCS is a key issue in these applications. In term of pose estimation, MCS can enhance the precision and robustness due to the redundant information they provided. Another advantage of MCS is that MCS can avoid the ambiguities in some single camera cases.

Although estimation of a single camera pose is a widely researched problem in the last decade, there are a limited number of dedicated methods for MCS pose estimation. In [13], Frahm et al. established a virtual camera based on the relation between two cameras, and converted the pose estimation of MSC into a problem of determining the pose of the virtual camera. Liu et al. retrieved the rotation and translation of MCS by optimizing the homographies of different cameras [2]. In [14], Xu et al. solved this problem by transferring all cameras into a unified coordinate and using the total error to replace the collinearity error in the well-known LHM algorithm [7].

In this paper, a novel robust binocular pose estimation method (which we refer to as BP*n*P) is proposed based on the RP*n*P method, and can be easily generalized to situations involving more than two cameras. Our method aims to obtain the 6 degrees of freedom (DOF) of a binocular camera system with an overlapping field-of-view by combining the observations of both cameras. We assume that each camera has known intrinsic parameters and fixed orientation and translation to each other which can be obtained by stereo calibration. In our method, we establish a unified coordinate frame to which both cameras refer, and then the pose for each camera can be calculated simultaneously and accurately.

In the RP*n*P method, a rotation axis is selected from a set of 3D reference points to establish a new orthogonal coordinate frame, which is an important factor that affects accuracy. The effects of rotation axis selection schemes have been discussed in [10], and selecting an edge whose projection length is the longest is reported to achieve better results than other schemes. But finding this edge from all the $n(n-1)/2$ ones in an *n*-point set is time-consuming, especially in large-size point sets. Furthermore, even though the one with the longest projection length is selected, it may not be the optimal rotation axis in this

978-1-4799-8389-6/15/$31.00 ©2015 IEEE

method. Therefore, the rotation axis selection step is formulated as an optimization problem in this paper, which can be solved efficiently by evolutionary algorithms.

Pigeon-inspired optimization (PIO) algorithm is a recently established evolutionary algorithm, which mimics the homing behavior of pigeons [17]. To apply this common natural phenomenon to a metaheuristic, Duan and Qiao designed two operators in this algorithm, i.e., map and compass operator, and landmark operator. PIO has demonstrated its competitive edge over several algorithms in real-world applications [17]-[19]. In this paper, an improved PIO algorithm is utilized to find the optimal rotation axis and therefore obtain the most accurate pose of the binocular camera system.

The rest of this paper is organized as follows. The details of our proposed BP*n*P method are presented in Section II. Section III describes the improved PIO algorithm and uses it to select the optimal rotation axis in our BP*n*P. Experimental results using synthetic data are given in Section IV. Finally, conclusions are drawn in Section V.

II. The BPnP Method

A. Problem Formulation

The problem of pose estimation for a calibrated binocular camera system can be defined as follows. The rotation and translation from the left camera to the right camera are denoted by $\begin{bmatrix} R^{rl} & t^{rl} \end{bmatrix}_{3\times4}$. Given several points $P_i\left(i=1,\ldots,n,n\geq3\right)$ in a 3D reference frame, their corresponding projections on normalized left image plane $p_i^l\left(i=1,\ldots,n\right)$ and right image plane $p_i^r\left(i=1,\ldots,n\right)$ (see Fig. 1), the task of this paper is to estimate the 6DOF of both the left and right camera which are denoted by $\begin{bmatrix} R^{l} & t^{l} \end{bmatrix}_{3\times4}$ and $\begin{bmatrix} R^{r} & t^{r} \end{bmatrix}_{3\times4}$, respectively. Since the rigid transformation between the two camera frames is known, once the 6DOF of the left camera are determined, those of the right camera can be obtained as follows:

$$\begin{bmatrix} R^r & t^r \\ 0^{\mathrm{T}} & 1 \end{bmatrix} = \begin{bmatrix} R^{rl} & t^{rl} \\ 0^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} R^l & t^l \\ 0^{\mathrm{T}} & 1 \end{bmatrix}. \tag{1}$$

Therefore, we define the left camera frame as the unified coordinate frame of the binocular camera system to make use of the information provided by both cameras. The binocular pose estimation problem is then converted into a problem of retrieving the rigid transformation between the unified coordinate frame and the reference frame.

B. Selection and Determination of a Rotation Axis

We begin with a selection of an edge $\overline{P_{i0}P_{j0}}$ from all the edges $\left\{\overline{P_iP_j} \mid i>j, i\in\left\{1,\ldots,n\right\}, j\in\left\{1,\ldots,n\right\}\right\}$ in the *n*-point set as a rotation axis. In [10], Li et al. provided three schemes to choose this edge and compared their influences on the accuracy of RP*n*P. This paper selects the rotation axis using an improved PIO algorithm, which will be described in Section III.

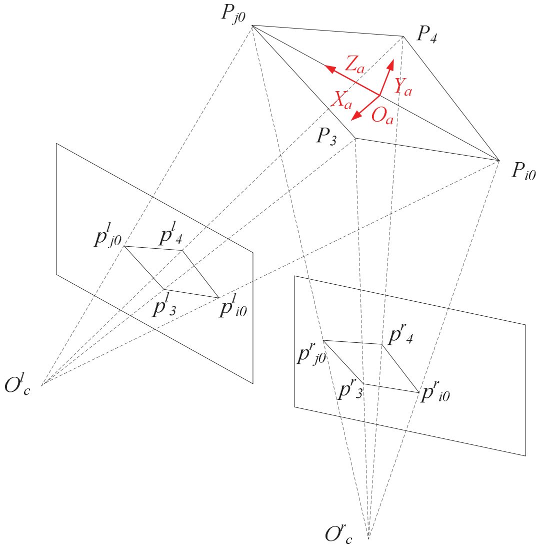


Fig. 1. The reference points and their projections in a binocular camera system.

A new orthogonal coordinate frame $O_aX_aY_aZ_a$ is established based on the selected edge $\overline{P_{i0}P_{j0}}$ as follows: The origin O_a is defined in the middle of this edge, and Z_a-axis has the same direction as $\overrightarrow{P_{i0}P_{j0}}$. After $O_aX_aY_aZ_a$ is created, the 3D points are transformed from original reference frame to the new coordinate frame.

In order to determine the rotation axis Z_a, the set with n points is divided into $\left(n-2\right)$ 3-point subsets. Applying 3-point constraint to the separate subset [20], we can get $\left(n-2\right)$ polynomials as follows:

$$\begin{cases} f_1\left(x\right)=\alpha_1x^4+\beta_1x^3+\chi_1x^2+\delta_1x+e_1=0, \\ f_2\left(x\right)=\alpha_2x^4+\beta_2x^3+\chi_2x^2+\delta_2x+e_2=0, \\ \ldots, \\ f_{n-2}\left(x\right)=\alpha_{n-2}x^4+\beta_{n-2}x^3+\chi_{n-2}x^2+\delta_{n-2}x+e_{n-2}=0. \end{cases} \tag{2}$$

The above nonlinear equation system can be solved by finding the minima of the square sum of the polynomials, the full derivation of which is given in [10]. When the minima are found, we can obtain the depths of P_{i0} and P_{j0}, based on which the rotation axis is determined:

$$Z_a=\frac{\overrightarrow{P_{i0}P_{j0}}}{\left\|\overline{P_{i0}P_{j0}}\right\|}. \tag{3}$$

Since each minimum corresponds to a rotation matrix and a translation vector, we choose the most accurate estimation as the final result.

C. Retrieving the Rotation and Translation

After determining the rotation axis, the rotation from the new coordinate frame to the left camera frame can be represented as follows:

$$R^l = Rrot\left(Z_a,\alpha\right) = \begin{bmatrix} r_1 & r_4 & r_7 \\ r_2 & r_5 & r_8 \\ r_3 & r_6 & r_9 \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

in which R is an orthogonal rotation matrix that has the same elements as Z_a in its third column.

Given *n* 3D reference points, the projections from them to the left and right image plane are formulated as follows:

$$\begin{cases} \lambda_i^l p_i^l = R^l P_i + t^l \\ \lambda_i^r p_i^r = R^{rl} R^l P_i + R^{rl} t^l + t^{rl} \end{cases}, \quad (5)$$

where $P_i = \left[X_i \quad Y_i \quad Z_i\right]^{\mathrm{T}}$ are the 3D coordinates of the reference point in $O_a X_a Y_a Z_a$, $p_i^l = \left(u_i, v_i, 1\right)^{\mathrm{T}}$ and $p_i^r = \left(u_i^r, v_i^r, 1\right)^{\mathrm{T}}$ are the normalized coordinates in the left and right image plane, respectively.

To retrieve the unknown variables $\cos\alpha$, $\sin\alpha$ and $t^l = \left[t_x^l \quad t_y^l \quad t_y^l\right]^{\mathrm{T}}$, we substitute (4) into (5) and rewrite it to a $4n \times 6$ homogenous linear equation system:

$$\left[A_{4n\times1}\ B_{4n\times1}\ C_{4n\times4}\right]\left[\cos\alpha \quad \sin\alpha \quad t_x^l \quad t_y^l \quad t_z^l \quad 1\right]^{\mathrm{T}} = 0, \quad (6)$$

where

$$A_{4n\times1} = \begin{bmatrix} u_1^l X_1 r_3 - Y_1 r_4 - X_1 r_1 + u_1^l Y_1 r_6 \\ v_1^l X_1 r_3 - Y_1 r_5 - X_1 r_2 + v_1^l Y_1 r_6 \\ u_1^r X_1 d_3 - Y_1 d_4 - X_1 d_1 + u_1^r Y_1 d_6 \\ v_1^r X_1 d_3 - Y_1 d_5 - X_1 d_2 + v_1^r Y_1 d_6 \\ \ldots \\ u_n^l X_n r_3 - Y_n r_4 - X_n r_1 + u_n^l Y_n r_6 \\ v_n^l X_n r_3 - Y_n r_5 - X_n r_2 + v_n^l Y_n r_6 \\ u_n^r X_n d_3 - Y_n d_4 - X_n d_1 + u_n^r Y_n d_6 \\ v_n^r X_n d_3 - Y_n d_5 - X_n d_2 + v_n^r Y_n d_6 \end{bmatrix},$$

$$B_{4n\times1} = \begin{bmatrix} Y_1 r_1 + u_1^l X_1 r_6 - u_1^l Y_1 r_3 - X_1 r_4 \\ Y_1 r_2 + v_1^l X_1 r_6 - v_1^l Y_1 r_3 - X_1 r_5 \\ Y_1 d_1 + u_1^r X_1 d_6 - u_1^r Y_1 d_3 - X_1 d_4 \\ Y_1 d_2 + v_1^r X_1 d_6 - v_1^r Y_1 d_3 - X_1 d_5 \\ \ldots \\ Y_n r_1 + u_n^l X_n r_6 - u_n^l Y_n r_3 - X_n r_4 \\ Y_n r_2 + v_n^l X_n r_6 - v_n^l Y_n r_3 - X_n r_5 \\ Y_n d_1 + u_n^r X_n d_6 - u_n^r Y_n d_3 - X_n d_4 \\ Y_n d_2 + v_n^r X_n d_6 - v_n^r Y_n d_3 - X_n d_5 \end{bmatrix},$$

$$C_{4n\times4} = \left[C_1 \quad C_2 \quad C_3 \quad C_4\right],$$

$$C_1 = \begin{bmatrix} -1 \\ 0 \\ u_1^r r_3^{rl} - r_1^{rl} \\ v_1^r r_3^{rl} - r_2^{rl} \\ \ldots \\ -1 \\ 0 \\ u_n^r r_3^{rl} - r_1^{rl} \\ v_n^r r_3^{rl} - r_2^{rl} \end{bmatrix}, \; C_2 = \begin{bmatrix} 0 \\ -1 \\ u_1^r r_6^{rl} - r_4^{rl} \\ v_1^r r_6^{rl} - r_5^{rl} \\ \ldots \\ 0 \\ -1 \\ u_n^r r_6^{rl} - r_4^{rl} \\ v_n^r r_6^{rl} - r_5^{rl} \end{bmatrix}, \; C_3 = \begin{bmatrix} u_1^l \\ v_1^l \\ u_1^r r_9^{rl} - r_7^{rl} \\ v_1^r r_9^{rl} - r_8^{rl} \\ \ldots \\ u_n^l \\ v_n^l \\ u_n^r r_9^{rl} - r_7^{rl} \\ v_n^r r_9^{rl} - r_8^{rl} \end{bmatrix},$$

$$C_4 = \begin{bmatrix} u_1^l r_9 Z_1 - r_7 Z_1 \\ v_1^l r_9 Z_1 - r_8 Z_1 \\ u_1^r d_9 Z_1 - d_7 Z_1 + u_1^r t_3^{rl} - t_1^{rl} \\ v_1^r d_9 Z_1 - d_8 Z_1 + v_1^r t_3^{rl} - t_2^{rl} \\ \ldots \\ u_n^l r_9 Z_n - r_7 Z_n \\ v_n^l r_9 Z_n - r_8 Z_n \\ u_n^r d_9 Z_n - d_7 Z_n + u_n^r t_3^{rl} - t_1^{rl} \\ v_n^r d_9 Z_n - d_8 Z_n + v_n^r t_3^{rl} - t_2^{rl} \end{bmatrix},$$

$$D = R^{rl} R = \begin{bmatrix} r_1^{rl} & r_4^{rl} & r_7^{rl} \\ r_2^{rl} & r_5^{rl} & r_8^{rl} \\ r_3^{rl} & r_6^{rl} & r_9^{rl} \end{bmatrix} \begin{bmatrix} r_1 & r_4 & r_7 \\ r_2 & r_5 & r_8 \\ r_3 & r_6 & r_9 \end{bmatrix} = \begin{bmatrix} d_1 & d_4 & d_7 \\ d_2 & d_5 & d_8 \\ d_3 & d_6 & d_9 \end{bmatrix}.$$

After solving this linear equation system, we can use the results to transform the 3D reference points into the left camera frame $O_c^l X_c^l Y_c^l Z_c^l$. Once the pose of the left camera is determined by 3D alignment scheme [10], that of the right camera can be obtained simultaneously according to (1).

III. PIO OPTIMIZED BPNP

In RP*n*P method, three schemes are employed to select the rotation axis, i.e., randomly choose an edge, choose an edge whose projection length is the longest from *n* randomly sampled edges and from all the $n\left(n-1\right)/2$ edges. The third scheme has been reported to be the most accurate. However, this scheme needs to calculate the projection lengths of all the $n\left(n-1\right)/2$ edges, which is not efficient, and the edge selected by it may not be the optimal one. Therefore, this paper formulates the rotation axis selection step as an optimization problem and solves it by an improved PIO algorithm.

In the PIO model, the position and velocity of the *i*th pigeon are denoted by $X_i = \left[x_{i1}, x_{i2}, \ldots, x_{im}\right]$ and $V_i = \left[v_{i1}, v_{i2}, \ldots, v_{im}\right]$, respectively, where *m* is the dimension of the problem to be solved. In this paper, $m = 2$ and $X_i = [x_{i1}, x_{i2}]$ represents the edge $\overline{P_{x_{i1}} P_{x_{i2}}}$. Therefore, the fitness value of this pigeon is the total error of BP*n*P when $\overline{P_{x_{i1}} P_{x_{i2}}}$ is selected as the rotation axis. The target of PIO algorithm here is to find the optimal rotation axis which yields the minimal total error.

The basic PIO model has two operators: the map and compass operator, and the landmark operator. The operator to be executed is randomly selected in each iteration. In the former operator, the position and velocity of the *i*th pigeon in the *t*th iteration are updated as follows:

$$\begin{cases} V_i(t)=V_i(t-1)\cdot e^{-Ft}+rand\cdot\left(X_g-X_i(t-1)\right) \\ X_i(t)=X_i(t-1)+V_i(t) \end{cases}, \tag{7}$$

where F is an important factor which can control the rate of the velocity change during iterations, *rand* is a random number generated within $[0,1]$, and X_g denotes the current global best position.

The population size of pigeons is decreased in the landmark operator, and the center position of the rest is given by:

$$X_c(t)=\frac{\sum_{N_p} X_i(t)\cdot\frac{1}{fitness\left(X_i(t)\right)}}{\sum_{N_p}\frac{1}{fitness\left(X_i(t)\right)}}, \tag{8}$$

where N_p is the current population size. Then the new position of each pigeon can be calculated by:

$$X_i(t)=X_i(t-1)+rand\cdot\left(X_c(t)-X_i(t-1)\right). \tag{9}$$

The reason why we use PIO algorithm instead of other evolution algorithms to select the rotation axis is that the convergence rate of PIO is so fast that an optimal result can be obtained within only a few iterations. But one drawback of the basic PIO algorithm is the lack of the diversity of population. In this paper, we improve the basic PIO algorithm by introducing a random binary integer-valued matrix (*map*) of size $N_p\times m$, which indicates the elements of $X_i(t)$ to be manipulated by using the corresponding elements of another item. For example, x_{ij} is updated only if $map_{ij}=1$. We modified (7) and (9) as follows:

$$\begin{cases} V_i(t)=V_i(t-1)\cdot e^{-Ft}+rand\cdot\left(X_g-X_i(t-1)\right) \\ X_i(t)=X_i(t-1)+map_i\cdot V_i(t) \end{cases}, \tag{10}$$

$$X_i(t)=X_i(t-1)+map_i\cdot rand\cdot\left(X_c(t)-X_i(t-1)\right), \tag{11}$$

where map_i is the *i*th row of the matrix *map*. The improved PIO algorithm is utilized to minimize the total error of BP*n*P method, the schematic diagram of which is shown in Fig. 2.

In addition, we have analyzed the computational complexity of PIO algorithm. Since the dimension of our problem is $n(n-1)/2$, i.e., the total number of edges from which the optimal rotation axis is to be picked out, the maximum number of iterations is denoted by N_c, and the size of the pigeon population is N_p, it can be calculated that the computational complexity of our method is $\Theta\left(n^2 N_c N_p\right)$. Therefore, the PIO optimized BP*n*P method can obtain the optimal solution within an acceptable period.

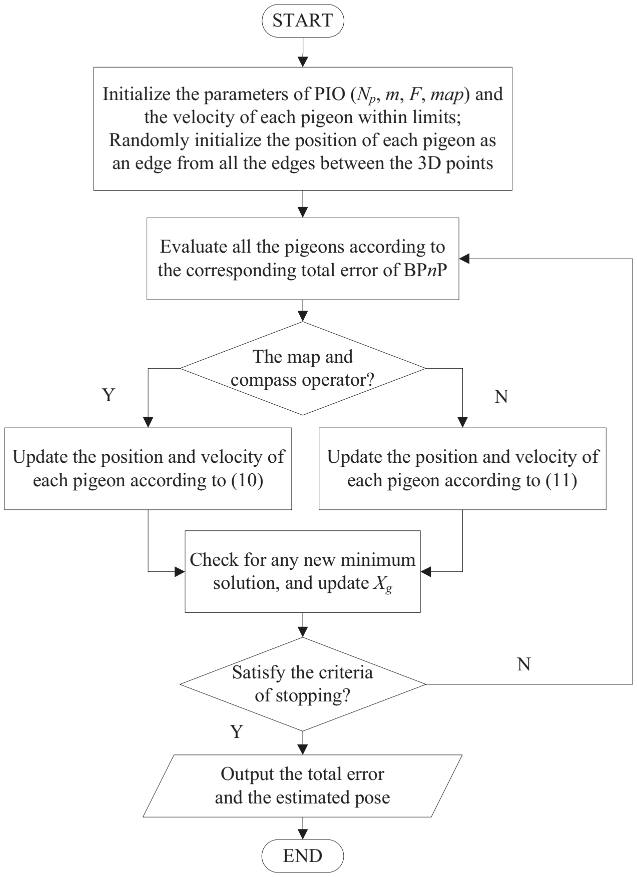

START
Initialize the parameters of PIO (N_p, m, F, *map*) and the velocity of each pigeon within limits; Randomly initialize the position of each pigeon as an edge from all the edges between the 3D points
Evaluate all the pigeons according to the corresponding total error of BP*n*P
The map and compass operator?
Y
N
Update the position and velocity of each pigeon according to (10)
Update the position and velocity of each pigeon according to (11)
Check for any new minimum solution, and update X_g
Satisfy the criteria of stopping?
N
Y
Output the total error and the estimated pose
END

Fig. 2. Schematic diagram of the PIO optimized BP*n*P method.

IV. Experiments

In this section, we compared the accuracy and robustness of our method (referred to as PIO-BP*n*P) against that of some state-of-the-art methods using synthetically generated data, which is the most prevalent way to evaluate the pose estimation methods.

The following algorithms were compared in the experiments:

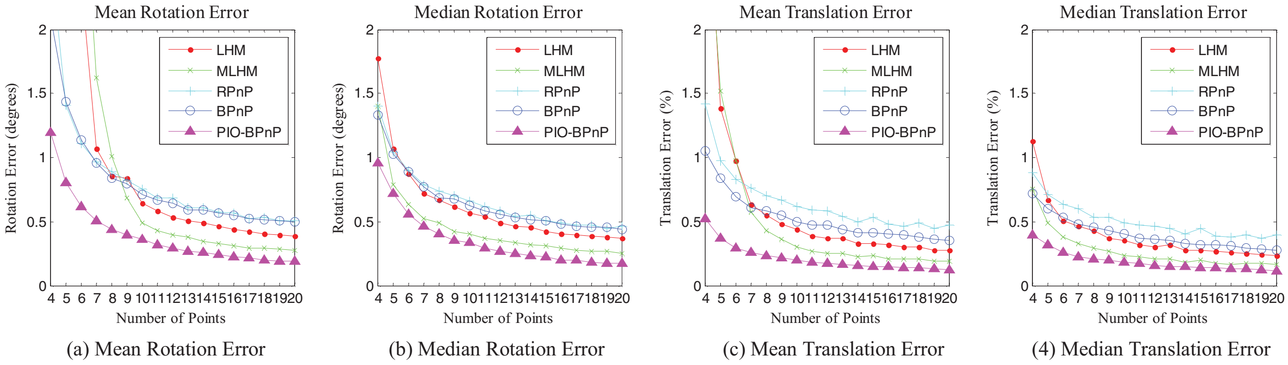
- LHM. A widely used iterative method for pose estimation by Lu et al [7], which is used to retrieve the pose of each camera individually.
- MLHM [14]. One of the best pose estimation methods for MCS, which is an enhancement of LHM. MLHM can obtain the 6DOF of a binocular camera system.
- RP*n*P. A very efficient and robust non-iterative P*n*P solution by Li et al [10], which can achieve as accurate results as the iterative methods. RP*n*P is used to determine the pose of each camera one by one.

- BP*n*P. The pose estimation method for binocular camera systems proposed in this paper, which can be used to estimate the poses of both cameras at the same time.
- PIO-BP*n*P. The PIO algorithm optimized BP*n*P method for estimating the pose of a binocular camera system, in which PIO is used to select the optimal rotation axis in BP*n*P.

The experimental configuration was as follows. We simulated two cameras with the same resolution 640×480 pixels, and the focal length of them was 800 pixels. The rotation and translation from the left camera to the right camera were given by:

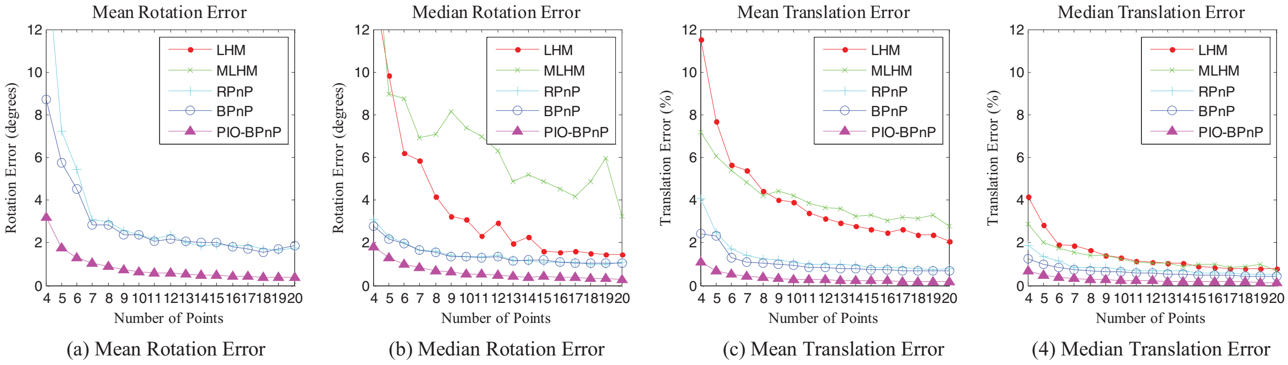
$$\begin{bmatrix} R^{rl} & t^{rl} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Three cases of the reference points were generated to evaluate these methods in our experiments, i.e., the ordinary 3D case, the planar case and the quasi-singular case, the definitions of which are given in [10]. In the three cases, the reference points were randomly distributed in the range $[-2,2]\times[-2,2]\times[4,8]$, $[-2,2]\times[-2,2]\times 0$ and $[1,2]\times[1,2]\times[4,8]$, respectively, and the same level ($\sigma = 3$) of Gaussian noise was added to their 2D projections on the left and right image plane. The pose of this binocular camera system was randomly generated.



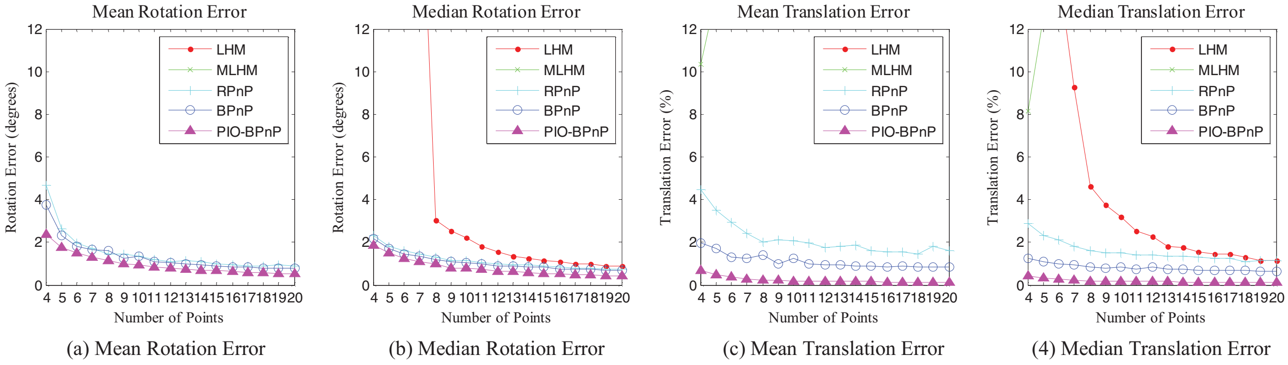
(a) Mean Rotation Error (b) Median Rotation Error (c) Mean Translation Error (4) Median Translation Error

Fig. 3. Results in the ordinary 3D case (the number of points $n = 4,...,20$, and the level of Gaussian noise $\sigma = 3$ pixels).



(a) Mean Rotation Error (b) Median Rotation Error (c) Mean Translation Error (4) Median Translation Error

Fig. 4. Results in the planar case (the number of points $n = 4,...,20$, and the level of Gaussian noise $\sigma = 3$ pixels).



(a) Mean Rotation Error (b) Median Rotation Error (c) Mean Translation Error (4) Median Translation Error

Fig. 5. Results in the quasi-singular case (the number of points $n = 4,...,20$, and the level of Gaussian noise $\sigma = 3$ pixels).

All experiments were performed in MATLAB R2014a on a 2.6 GHz Intel Corel i5 processor. The parameters of PIO algorithm were set as: $N_p = 30$, $m = 2$, $F = 1$, and the maximum number of iterations $N_c = 30$. Each experiment was carried out for 1000 times and the average result was reported. The results of the three cases are given from Fig. 3 to Fig. 5.

According to Fig. 3, in the ordinary 3D case, all of the five methods can reach very accurate results. Both of the RP*n*P and BP*n*P method can achieve better performances than the iterative algorithm LHM when only a small number of points are available. MLHM and PIO-BP*n*P are the two best algorithms in this case, in which our PIO-BP*n*P achieves the most correct results from 4 points to 20 points.

In the planar case shown in Fig. 4, LHM and MLHM are not stable for estimating the rotation. The mean rotation errors of LHM and MLHM are higher than 24 degree and 46 degree, respectively, which are outside the reported range in Fig 4. The accuracy of BP*n*P is similar as that of RP*n*P in this case, and both of them are much better than LHM and MLHM. As for our PIO-BP*n*P method, the errors of both translation and rotation are considerably lower than those of other algorithms.

As can be seen in Fig. 5, LHM and MLHM cannot work well in the quasi-singular case because of the local minima problem. Although RP*n*P can still achieve relatively accurate results in this case, the translation results of it are worse than those obtained by BP*n*P. Due to the rotation axis selection step optimized by PIO algorithm, PIO-BP*n*P can stably achieve the most accurate pose results for binocular camera systems in all the case.

V. Conclusion

This paper presented a robust and accurate pose estimation algorithm for binocular camera systems, which can be also used to estimate the pose of MCS. In the experimental section, we have shown that the proposed BP*n*P method is capable of dealing with different configurations of test data sets. An improved PIO algorithm is utilized to optimize the rotation axis selection step in our BP*n*P method, and it achieves much more accurate results than other state-of-the-art methods in all tested cases.

Acknowledgment

The authors would like to thank S. Li, C. Xu, and M. Xie for their excellent work on RP*n*P. This work was partially supported by National Natural Science Foundation of China under grant #61425008, #61333004 and #61273054, Aeronautical Foundation of China under grant #20135851042, and Graduate Innovation Foundation for Beihang University under Grant #YCSJ-01-2014-01.

References

[1] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge, MA: Cambridge University Press, 2000.

[2] Y. Yu, K. Wong, and M. Chang, “Pose estimation for augmented reality applications using genetic algorithm,” IEEE Trans. Syst., Man, Cybern.–Part B: Cybern.,vol. 35, no. 6, pp. 1295-1301, December 2005.

[3] S. Li and C. Xu, “Efficient lookup table based camera pose estimation for augmented reality,” Comput. Anim. Virtual Worlds, vol. 22, no. 22, pp. 47-58, January 2011.

[4] Q. Ji, M. Costa, R. Haralick, and L. Shapiro, “A robust linear least-squares estimation of camera exterior orientation using multiple geometric features,” ISPRS J. Photogramm. Remote Sens., vol. 55, no. 2, pp. 75–93, June 2000.

[5] Y. Liu, R. Xiong, and Y. Li, “Robust and accurate multiple-camera pose estimation toward robotic applications,” Int. J. Adv. Rob. Syst., vol. 11, no. 153, pp. 1–13, September 2014.

[6] H. Duan and Q. Zhang, “Visual measurement in simulation environment for vision-based UAV autonomous aerial refueling,” IEEE Trans. Instrum. Meas., in press.

[7] C. Lu, G. Hager, and E. Mjolsness, “Fast and globally convergent pose estimation from video images,” IEEE Trans. Pattern. Anal. Mach. Intell., vol. 22, no. 6, pp. 610–622, June 2000.

[8] G. Schweighofer and A. Pinz, “Robust pose estimation from a planar target,” IEEE Trans. Pattern. Anal. Mach. Intell., vol. 28, no. 12, pp. 2024–2030, December 2006.

[9] Z. Zhang, “A flexible new technique for camera calibration,” IEEE Trans. Pattern. Anal. Mach. Intell., vol. 22, no. 11, pp. 1330–1334, November 2000.

[10] S. Li, C. Xu, and M. Xie, “A robust $\Theta(n)$ solution to the perspective-*n*-point problem,” IEEE Trans. Pattern. Anal. Mach. Intell., vol. 34, no. 7, pp. 1444–1450, July 2012.

[11] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EP*n*P: An accurate O(n) solution to the P*n*P problem,” Int. J. Comput. Vision, vol. 81, no. 2, pp. 155–166, February 2008.

[12] A. Ansar and K. Daniilidis, “Linear pose estimation from points or lines,” IEEE Trans. Pattern. Anal. Mach. Intell., vol. 25, no. 5, pp. 578–589, May 2003.

[13] J. Frahm, K. Köser, and R. Koch, “Pose estimation for multi-camera systems,” in Pattern Recognition. Springer Berlin Heidelberg, 2004, pp. 286–293.

[14] Y. Xu, Y. Jiang, and F. Chen, “Generalized orthogonal iterative algorithm for pose estimation of multiple camera systems,” Acta Optica Sinica, vol. 29, no. 1, pp. 72–77, January 2009.

[15] C. Chen and D. Schonfeld, “Robust 3D pose estimation from multiple video cameras,” in Proc. Int. Conf. Image Process. ICIP, pp. 541–544, November 2009.

[16] G. Lee, M. Pollefeys, and F. Fraundorfer, “Relative pose estimation for a multi-camera system with known vertical direction,” in Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit., pp. 540–547, September 2014.

[17] H. Duan and P. Qiao, “Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning,” Int. J. Intell. Comput. Cybern., vol. 7, no. 1, pp. 24–37, March 2014.

[18] C. Li and H. Duan, “Target detection approach for UAVs via improved pigeon-inspired optimization and edge potential function,” Aerosp. Sci. Technol., vol. 39, pp. 352–360, December 2014.

[19] R. Dou and H. Duan, “Pigeon inspired optimization approach to model prediction control for unmanned air vehicle,” Aircr. Eng. Aerosp. Technol., in press.

[20] S. Li and C. Xu, “A stable direct solution of perspective-three-point problem,” Int. J. Pattern Recognit. Artif. Intell., vol. 25, no. 5, pp. 627–642, August 2011.