International Journal of Advanced Robotic Systems

OPEN ACCESS
ARTICLE

Robust and Accurate Multiple-camera Pose Estimation Toward Robotic Applications

Regular Paper

Yong Liu[1,2], Rong Xiong[1,2,*] and Yi Li[1]

1 State Key Laboratory of Industrial Contol Technology, Zhejiang University, Hangzhou, China
2 Institue of Cyber-Systems and Control, Zhejiang University, Hangzhou, China
* Corresponding author E-mail: rxiong@iipc.zju.edu.cn

Received 29 Apr 2014; Accepted 09 Jul 2014

DOI: 10.5772/58868

Abstract Pose estimation methods in robotics applications frequently suffer from inaccuracy due to a lack of correspondence and real-time constraints, and instability from a wide range of viewpoints, etc. In this paper, we present a novel approach for estimating the poses of all the cameras in a multi-camera system in which each camera is placed rigidly using only a few coplanar points simultaneously. Instead of solving the orientation and translation for the multi-camera system from the overlapping point correspondences among all the cameras directly, we employ homography, which can map image points with 3D coplanar-referenced points. In our method, we first establish the corresponding relations between each camera by their Euclidean geometries and optimize the homographies of the cameras; then, we solve the orientation and translation for the optimal homographies. The results from simulations and real case experiments show that our approach is accurate and robust for implementation in robotics applications. Finally, a practical implementation in a ping-pong robot is described in order to confirm the validity of our approach.

Keywords Multi-camera System, Pose Estimation, Coplanar Points, Ping-pong Robot

1. Introduction

Pose estimation from a referenced rigid structure is one of the most basic and important problems in robotics vision and computer graphics. It can be used to obtain the 6DOF (degrees of freedom) of the cameras needed for further implementation, e.g., in helping robots to locate targets in referenced coordinates [1, 2], in calculating coordinates in images of virtual 3D objects to synthesize augmented reality scenes [3], in locating flying balls for robots [4], and in calibrating the camera-laser sensor system [5].

Although the single camera pose estimation problem has been widely researched in the last decade, the methods used often suffer from low robustness and ill-conditioned pose estimation problems. Furthermore, a relatively small angle for viewing influences the accuracy of the estimated pose.

In practical robotics vision applications (e.g., catching objects via robot arms, playing ping-pong with robots, etc., all of which may require the vision system to perform in real-time), the tasks of pose estimation (which aim to locate the absolute coordinates of objects via the vision system of the robot) require quite a high degree of accuracy and robust performance in most conditions. A single camera cannot provide enough precision or robustness in such

tasks for pose estimation, and thus a multiple camera system is required. Furthermore, task-related robots are normally required to obtain the absolute coordinates of a tracing target in the referenced space via rigid point correspondences from landmarks, while a single-camera system will obtain infinite solutions for the target due to the property of perspective projection.

In the specific case of mobile robots, pose estimation methods for multiple camera systems have always faced the following challenges:

(1) In applications of mobile robots - especially in the case of robots working at high speed - to achieve the poses of all the cameras simultaneously is the most important requirement. Otherwise, the poses estimated individually will break the rigid constraint among all the cameras in the multiple camera system, as there may be some motion in the pose estimation of different cameras due to the high speed of the robot.

(2) Methods are always required that can work accurately, stably (with low standard deviation - STD) and robustly under any angle of viewpoint due to the uncertain poses of mobile robots.

(3) The task of localization is always followed by the task of pose estimation; thus, the methods should also provide preferable localization performance for their estimations.

(4) As the movement of the robot may shake the rigid rig among the cameras and introduce bias, the pose estimation methods should also be robust in relation to the interference of bias on solid rigs.

(5) Mobile robots are real-time systems, and so they require the pose estimation methods to calculate as quickly as possible and with fewer point correspondences, due to the mobility of a given view.

In this paper, we address all the above challenges and present a novel approach for estimating the poses of all the cameras in a multi-camera system with only a few coplanar points in a manner that is accurate, robust and simultaneous[1], and we aim to resolve the above four main challenges as they arise in the practical vision system of a humanoid ping-pong robot.

2. Related Works

Pose estimation for single cameras [6–11] has been studied for many years. Recently, much of the research has focused on pose estimation in multi-camera systems, due to the limitations of single cameras, e.g., their low accuracy and limited field-of-view.

One of the most important advantages of a multi-camera system is that it can recover stereo information easily (e.g., the visual odometry [12], which employs calibrated cameras to recover the 3D information of targets), and it can help to estimate the motion of the cameras from the optical flow via the Kalman filter method or by minimizing a cost function based on the geometric and 3D properties of the features.

Generally, there are two kinds of multi-camera systems: one is designed for overlapping fields of view [13–17] and the other for non-overlapping fields of view [18, 19].

The methods for a non-overlapping system [18, 19] require the use of cameras which are placed rigidly on a moving object, where the translation and rotation between the cameras are known. When the object is moving rigidly, these methods can recover the 6DOF motion for the multi-camera system via the point correspondences between two points seen before and after motion. These methods are normally implemented on a vehicle or other moving objects, and require relative motion between every other image.

The methods for overlapping systems also employ cameras placed rigidly, with known translation and rotation between each camera, and they can recover the pose of their systems with only one frame of the multiple images from different cameras. As such, these methods can be used to process static scenes. As there are many efficient pose estimation methods for single cameras, an intuitive solution for overlapping fields of view in multi-camera systems is to estimate the pose for each camera and then reduce the ambiguities produced by the estimated poses of every camera based on their rigid constraint via fusing or polling policies. The methods presented by Baker et al. [14] and Viksté et al. [15] belong to this category. However, this kind of method does not obtain a unifying pose for the multi-camera system with all the information from all the cameras simultaneously. It may introduce some inconsistencies with the rigid constraint between each pair of cameras, which may reduce the precision of multi-camera systems for measurement or object localization (i.e., stereo vision for grasping with robots [20]). In this paper, we present a novel approach which estimates the pose of a multi-camera system with overlapping fields of view and can calculate the unifying pose for the system with all the information from all the cameras simultaneously.

3. Our Approach to Pose Estimation for Multi-camera Systems

Pose estimation for a multi-camera system should calculate the orientation and translation with the rigid pose constraint among the various cameras. Most existing methods attempt to solve the orientation and translation directly, by optimization [21] or iteration [11, 22]. In contrast, we employ homography, which is widely implemented in calibration [23, 24] and can map image points with 3D coplanar referenced points. Firstly, we establish the corresponding relations between each camera using their Euclidean geometries and optimize the homographies of the cameras; then, we solve the orientation and translation from the optimal homographies.

[1] Here, we only need to estimate one representative pose of the multi-camera system - any other poses of any other camera may be obtained by transforming with the known orientations and translations existing between each camera.

3.1 Problem definitions

The intrinsic parameters of the i^{th} camera in a multi-camera system are denoted by $\mathbf{K_j}$. The spatial 3D point $\mathbf{M} = [X \quad Y \quad Z]^T$ has its image on the jth camera, denoted by $\mathbf{m^j} = [u^j \quad v^j]^T$. The absolute 6DOF[2](three for rotation and three for translation) of this camera are denoted by $[\mathbf{R_j} \quad \mathbf{t_j}]_{3\times4}$. The rotation and translation from camera j to camera i are denoted by $[\mathbf{R_{ij}} \quad \mathbf{t_{ij}}]_{3\times4}$, which can be calibrated accurately as previously, since multi-camera systems are rigid. The translation between cameras i and j can be calculated as follows:

$$\begin{bmatrix} \mathbf{R_i} & \mathbf{t_i} \\ \mathbf{0^T} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R_{ij}} & \mathbf{t_{ij}} \\ \mathbf{0^T} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R_j} & \mathbf{t_j} \\ \mathbf{0^T} & 1 \end{bmatrix} \tag{1}$$

The task of estimating the absolute pose of a multi-camera system can be formalized as follows. Given a set of coplanar non-collinear 3D coordinates of referenced points $\mathbf{M_i}, i = 1, ..., n, n \geq 3$, the corresponding images in a multi-camera system are $\mathbf{m_i^j}$, $i = 1, ..., n, n \geq 3$ and $j = 1, ..., k, k \geq 2$, which is the number of cameras. Based on the above corresponding points, it will estimate the $\mathbf{R_j}$, $\mathbf{t_j}$ for each camera with the known rigid translation between cameras $[\mathbf{R_{ij}} \quad \mathbf{t_{ij}}]$, $i, j = 1, ..., k$.

3.2 Homography for coplanar corresponding points

The homogeneous coordinate of the referenced point $\tilde{\mathbf{M}} = [X \quad Y \quad Z \quad 1]^T$ and the corresponding image point of camera j in the homogeneous coordinate $\tilde{\mathbf{m}} = [u \quad v \quad 1]^T$ has a project relation, and here the 3D referenced points are coplanar. As such, we can assume that $Z = 0$, and we have:

$$\begin{aligned} s\tilde{\mathbf{m}} = \mathbf{K_j}[\mathbf{R_j} \quad \mathbf{t_j}]\tilde{\mathbf{M}} &= \mathbf{K_j}[\mathbf{r_{1_j}} \quad \mathbf{r_{2_j}} \quad \mathbf{r_{3_j}} \quad \mathbf{t_j}]\tilde{\mathbf{M}} \\ &= \mathbf{K_j}[\mathbf{r_{1_j}} \quad \mathbf{r_{2_j}} \quad \mathbf{t_j}][X \quad Y \quad 1]^T \end{aligned} \tag{2}$$

Here, s is an arbitrary scale factor, $\mathbf{K_j}$ are the intrinsic parameters of the j^{th} camera, and $\mathbf{R_j}$ and $\mathbf{t_j}$ are the rotation and translation, respectively, of the corresponding camera j. In the remainder of this paper, we use $\tilde{\mathbf{M}}$ to represent the homogeneous coordinate of $\mathbf{M}$, even though $\tilde{\mathbf{M}} = [X \quad Y \quad 1]^T$. Next, we introduce the homography $\mathbf{H}$ within Zhang's calibration method [24]:

$$\mathbf{H_j} = \lambda_j \mathbf{K_j}[\mathbf{r_{1_j}} \quad \mathbf{r_{2_j}} \quad \mathbf{t_j}] \text{ with } s_j\tilde{\mathbf{m}} = \mathbf{H_j}\tilde{\mathbf{M}} \tag{3}$$

$\mathbf{H}$ is a 3×3 matrix, also defined upon a scale factor s_j. For convenience of calculation, here, we only define the scale factor with $H(3,3) = 1$. Assuming that $\mathbf{H_j} = [\mathbf{h_{1_j}} \quad \mathbf{h_{2_j}} \quad \mathbf{h_{3_j}}]$, then from equation (3) the rotation and translation vector can be calculated as follows:

$$\begin{aligned} \mathbf{r_{1_j}} &= \lambda_j \mathbf{K_j^{-1}} \mathbf{h_{1_j}} \\ \mathbf{r_{2_j}} &= \lambda_j \mathbf{K_j^{-1}} \mathbf{h_{2_j}} \\ \mathbf{r_{3_j}} &= \mathbf{r_{1_j}} \times \mathbf{r_{2_j}} \\ \mathbf{t_j} &= \lambda_j \mathbf{K_j^{-1}} \mathbf{h_{3_j}} \end{aligned} \tag{4}$$

$\lambda_j = 1/ \parallel \mathbf{K_j^{-1}} \mathbf{h_{1_j}} \parallel = 1/ \parallel \mathbf{K_j^{-1}} \mathbf{h_{2_j}} \parallel$, and so the aim of estimating the 6DOF pose of the camera can be switched to estimating the homography of the corresponding camera.

3.3 The global optimum for multi-camera systems

In our approach, we try to minimize the image distances of all the cameras with respect to the referenced coplanar points.

Given n referenced coplanar points and k cameras, we can estimate the homography ($\mathbf{H_j}$, $j = 1, ..., k$) for each camera by minimizing the following function:

$$\sum_{j=1}^{k} \sum_{i=1}^{n} \parallel \tilde{\mathbf{m}}_\mathbf{i}^\mathbf{j} - \frac{1}{s_j} \mathbf{H_j} \tilde{\mathbf{M}}_\mathbf{i} \parallel^2 \tag{5}$$

The above optimal function is established on the assumption that the image points of each camera are perturbed by Gaussian noise, which is quite usual in many image noise removal methods [25–27] and vision practices [28–30]. For cameras that are assembled rigidly, the homography of each camera has inherent constraint relations, which will help us to obtain the global optimization for the multi-camera system using only a few points.

3.4 Extrinsic translation for multiple cameras with homography

In this section, we will present the method for the calculation a camera's homography from one known homography with known rigid rotation and translation. Assume that there are two cameras, i and j, with a rigid rotation matrix $\mathbf{R_{ij}}$ and translation vector $\mathbf{t_{ij}}$. Based on equations (1) and (4), the translation and rotation between these two cameras are:

$$\mathbf{t_i} = \mathbf{R_{ij}} * \mathbf{t_j} + \mathbf{t_{ij}} = \lambda_j \mathbf{R_{ij}} \mathbf{K_j^{-1}} \mathbf{h_{3_j}} + \mathbf{t_{ij}} \tag{6}$$

$$\mathbf{R_i} = \mathbf{R_{ij}} * [\lambda_j \mathbf{K_j^{-1}} \mathbf{h_{1_j}} \quad \lambda_j \mathbf{K_j^{-1}} \mathbf{h_{2_j}} \quad \lambda_j \mathbf{K_j^{-1}} \mathbf{h_{1_j}} \times \lambda_j \mathbf{K_j^{-1}} \mathbf{h_{2_j}}] \tag{7}$$

Thus, the homography of camera i has the following relation based on the definition of homography in equation (3):

$$\mathbf{H_i} \sim \mathbf{H_i'} = \mathbf{K_i}[\lambda_j \mathbf{R_{ij}} \mathbf{K_j^{-1}} \mathbf{h_{1_j}} \quad \lambda_j \mathbf{R_{ij}} \mathbf{K_j^{-1}} \mathbf{h_{2_j}} \quad \lambda_j \mathbf{R_{ij}} \mathbf{K_j^{-1}} \mathbf{h_{3_j}} + \mathbf{t_{ij}}] \tag{8}$$

The above formulation presents only the scale transformation relation between the homographies of two different cameras. However, in our approach, the homography is defined with the scale $H(3,3) = 1$, and

[2] Most approaches provide only 5DOF motion, and the scale of translation cannot be recovered. As such, we call our method 'absolute pose estimation'.

so the homography of $\mathbf{H_i}$ can be calculated using the function and input parameters $\mathbf{H_j}$, $\mathbf{R_{ij}}$, $\mathbf{t_{ij}}$:

$$\check{H}_i(\mathbf{H_j}, \mathbf{R_{ij}}, \mathbf{t_{ij}}) = \frac{\mathbf{H_i'}}{H_i'(3,3)} \tag{9}$$

With homography $\mathbf{H_i}$, the rotation and translation vector for camera i can be easily calculated by the formula (4).

Accordingly, the optimization function in (5) can be rewritten as:

$$\arg\min_{\mathbf{H_q}} \sum_{j=1}^{k} \sum_{i=1}^{n} \parallel \tilde{\mathbf{m}}_\mathbf{i}^\mathbf{j} - \frac{1}{s_j} \check{H}_j(\mathbf{H_q}, \mathbf{R_{jq}}, \mathbf{t_{jq}}) \tilde{\mathbf{M}}_\mathbf{i} \parallel^2 \tag{10}$$

Here, the scale factor s_j can be calculated with the following formula:

$$\frac{1}{s_j} \mathbf{H_j} \tilde{\mathbf{M}}_\mathbf{i} = \begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \\ 1 \end{bmatrix} \tag{11}$$

3.5 Estimation of the initial homography

The solution for the formula (10) is a typical nonlinear optimization problem. In our approach, the Levenberg-Marquardt method is adopted, which has been widely used in computer vision cases. When solving the optimization with formula (10), an initial guess for $\mathbf{H_q}$ is needed.

The method to calculate the initial homography $\mathbf{H}$ is similar to Zhang's approach [24] to calibration. Firstly, assuming that $\mathbf{x} = [\bar{\mathbf{h}}_\mathbf{1}^\mathbf{T} \quad \bar{\mathbf{h}}_\mathbf{2}^\mathbf{T} \quad \bar{\mathbf{h}}_\mathbf{3}^\mathbf{T}]^T$, and given equation 3, we have:

$$\begin{bmatrix} \tilde{\mathbf{M}}^\mathbf{T} & 0^T & -u\tilde{\mathbf{M}}^\mathbf{T} \\ 0^T & \tilde{\mathbf{M}}^\mathbf{T} & -v\tilde{\mathbf{M}}^\mathbf{T} \end{bmatrix} \mathbf{x} = 0 \tag{12}$$

Since $\mathbf{x}$ is defined using a scale factor, the solution for the above equation can be implemented with a singular decomposition, and we can rewrite equation (12) as $\mathbf{Ax} = 0$ (here, $\mathbf{A}$ is a $2n \times 9$ matrix and n is the number of referenced points) and obtain the correct singular vector of $\mathbf{A}$ associated with the smallest singular value [24].

Before calculating the initial guess as to the homography using the above method, the data should first be normalized [28] to obtain more stable and accurate results.

If a more accurate initial guess for the homography is desired, the maximum likelihood estimation of $\mathbf{H}$ can also be applied using the following function:

$$\sum_{i=1}^{n} \parallel \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \check{\mathbf{m}}_\mathbf{i} \parallel^2 \tag{13}$$

Such that $\check{\mathbf{m}}_\mathbf{i} = \frac{1}{\bar{\mathbf{h}}_\mathbf{3}^\mathbf{T} \tilde{\mathbf{M}}_\mathbf{i}} \begin{bmatrix} \bar{\mathbf{h}}_\mathbf{1}^\mathbf{T} \tilde{\mathbf{M}}_\mathbf{i} \\ \bar{\mathbf{h}}_\mathbf{2}^\mathbf{T} \tilde{\mathbf{M}}_\mathbf{i} \end{bmatrix}$, with $\bar{\mathbf{h}}_\mathbf{i}^\mathbf{T}$, is the ith row of $\mathbf{H}$. The optimization of the above function can also be solved using the Levenberg-Marquardt method, and the initial guess can be obtained with the solution to equation (12).

4. Experiments

We compared our method with various state-of-the-art methods, using both simulations and real image data. Finally, we present the practical implementation of our method in ping-pong robots.

In the following experiments, we used seven pose estimation methods.

- *MAAMC*, the method introduced in this paper (abbreviated as *MAAMC*) to obtain the pose of the multi-camera system simultaneously.
- *SAAMC*, a simplified version of our *MAAMC* - it separates the computation of the homography for each camera and then obtains the pose of each camera individually with formula (3).
- *LHM*, the method presented by Lu *et al.*. [11], which is an efficient iterative approach for estimating the pose for a single camera. This method has been shown to be capable of dealing with arbitrary numbers of correspondences and achieves excellent precision when it converges properly[3]. We used *LHM* to estimate the pose of each camera individually.
- *MLHM* [17], an enhancement of LHM, which transfers all the cameras into one unified coordinate and minimizes the total errors in 3D referenced space iteratively. This method can also obtain the 6DOF pose of the multi-camera system simultaneously. In our experiments, weak-perspective models were used to obtain the initial estimations for both *LHM* and *MLHM*.
- *JKR* [16], a method which can estimate the poses of each camera simultaneously.
- *RPP* [10], a robust pose estimation method from a planar target presented by Schweighofer. It estimates each camera's pose individually.
- $MAACM - MLHM$, a hybrid method which employs our initial pose estimation method to obtain the initial estimation for *MLHM*.

The above seven methods could be classified according to three categories. One set includes the methods [4] estimated from coplanar points, e.g., *MAAMC*, *SAAMC* and *RPP*. The second set includes the methods [5] estimated from non-coplanar points, e.g., LHM and MLHM. The last method is $MAACM - MLHM$ [6], which is a premium version of *MLHM* with an initial R estimated by *MAACM*.

With the above seven methods, *MLHM*,*MAAMC*,*JKR* and $MAAMC - MLHM$ were all able to obtain the poses of each camera simultaneously, while the other approaches needed to estimate the pose of each camera one by one.

[3] In our experiments, the code of *LHM* was downloaded from the author's website, http://www.cs.jhu.edu/∼hager (30/04/2014)

[4] In the following experiments, we use red lines to represent those pose estimation methods from coplanar points.

[5] In the following experiments, we use green lines to represent those pose estimation methods from non-coplanar points.

[6] In the following experiments, we use blue lines to represent this method.

4.1 Simulation experiments

In these experiments, we simulated several cameras placed rigidly. The distances [7] between cameras were about 100. Each camera's focal length ratio was set as $f_u = f_v = 540$ and the simulation resolution was 640×480; thus, the principal point was located at the pixel point (U_0, V_0)=(320,240). The Gaussian noise for the corresponding 2D point coordinates was also included in the simulation model.

We used the relative error to evaluate the experimental results. Given the true results of camera i, $\tilde{\mathbf{R}}_\mathbf{i}$, $\tilde{\mathbf{t}}_\mathbf{i}$, the relative errors for the multi-camera system are defined as follows:

$$E_{rot}(\%) = \frac{1}{n}\sum_{i=1}^{n}\frac{\parallel \tilde{\mathbf{R}}_\mathbf{i} - \mathbf{R}_\mathbf{i} \parallel}{\parallel \mathbf{R}_\mathbf{i} \parallel} \tag{14}$$

$$E_{tran}(\%) = \frac{1}{n}\sum_{i=1}^{n}\frac{\parallel \tilde{\mathbf{t}}_\mathbf{i} - \mathbf{t}_\mathbf{i} \parallel}{\parallel \mathbf{t}_\mathbf{i} \parallel} \tag{15}$$

where n is the total number of cameras in the system, and $\mathbf{R}_\mathbf{i}$ and $\mathbf{t}_\mathbf{i}$ are camera i's rotation and translation, respectively, obtained by the pose estimation approach. We executed each experiment 300 times independently in MATLAB and recorded the average.

To evaluate the effect of the estimation when the two cameras are placed and subject to different amounts of rigid motion, all the simulation experiments were designed with random relative positions of the camera, which was achieved by limiting three axis-rotation angles with intervals of [0,2] degrees, translation vectors from [50,0,0] to [60,10,10], and while controlling all the reference points in the overlapping view.

4.1.1. Simulation experiments for R, T, localization and re-projection error

As the performance of the pose estimation methods may be influenced by the viewing angle, we evaluated the performance (R, T, localization and image re-projection error) of the above methods given small viewing angles (which occurred quite frequently in the humanoid ping-pong robot).

In the simulation experiment, the 3D referenced points were restricted in the plane with $z = 0$, $x \in [-200, 200]$ and $y \in [-300, 300]$. The cameras were placed within the conic area (the vertex was $(0, 0, 0)$), where the distribution intervals were $z \in [1500, 1700]$, $x \in [-980, 980]$ and $y \in [2500, 3200]$. The experimental results are shown in figure 1 and figure 2.

We also carried out an experiment to evaluate the absolute locating performances of these methods [8]. When executing the pose estimation, a noised point (with a ground true value $P(430, 760, 200)$) was generated out of the plane of referenced points, and we used the estimated pose of each camera to estimate the coordinate of point P with its image (with the same Gauss noise as the referenced points) in each camera. Its position in the referenced coordinates was calculated by taking a radial from the centre of the projection of each camera to the image of P in that camera, and calculating the middle points of the perpendicular bisectors between every two radials. The estimated $\tilde{P}$ was then obtained by averaging all the middle points. The experimental results from the locating method are shown in figure 1 and figure 2 (e) and (g).

According to figure 1(a),(b),(c) and (d), we can see that the object space-minimizing-based approaches, i.e., $MLHM$ and LHM, will be much worse than the other approaches for small angle of viewing conditions with increasing noise. The reason may be that the initial guesses of $MLHM$ and LHM are normally far from the true pose when the viewing angles are small; thus, they will converge on another local minimum. That is why $MAACM - MLHM$ will achieve a better performance with a better initial guess from $MAACM$. With such small angle viewing conditions, RPP has a special optimization to avoid converging on the local minimum so that it can achieve the same best performance as our $SAACM$ approach for pose estimation.

According to figure 2(a)-(d), we can see that the object space-minimizing-based approaches, i.e., $MLHM$, LHM will be quite unstable for pose estimation for small viewing angles. Figure 2(e) and (g) also show that the approaches estimating poses simultaneously will achieve better performances than those doing so individually for localization.

The simulation experimental results also show that the pose estimation for each camera in turn could perform as well as the global optimization methods; however, the performances in localization with those methods estimated one by one were always worse than the global methods. In our view, the reason for this may be that the global optimization methods always consider the rigid rig as the constraint in optimization, which may lead to greater compromise in the computation of the average system bias. However, the method optimizing singly can ignore this constraint and converge on those poses with less system bias, while the real rig is broken and leads to poor performance in localization applications.

4.1.2. Simulation Experiments on Other Metrics

In the second simulation experiment, we further evaluated four approaches, i.e., JKR, $MLHM$, $MAAMC$ and $MAAMC - MLHM$, all which estimated the poses of all the cameras simultaneously.

We first evaluated the performances of the four approaches for different mounted cameras. Figure 3 shows the experimental results after using various numbers of cameras (3-5) in the multi-camera system. In this experiment, we randomly chose 25 poses for the multi-camera system, which were distributed in the half

[7] In the following experiments, the unit for distance for a coordinate is mm.

[8] In this experiment, we employed a very simple location method that extracts lines from each camera with two points - the image point of the object and the optic centre-point of each camera - and then calculated the centre of those lines as the position of the target.

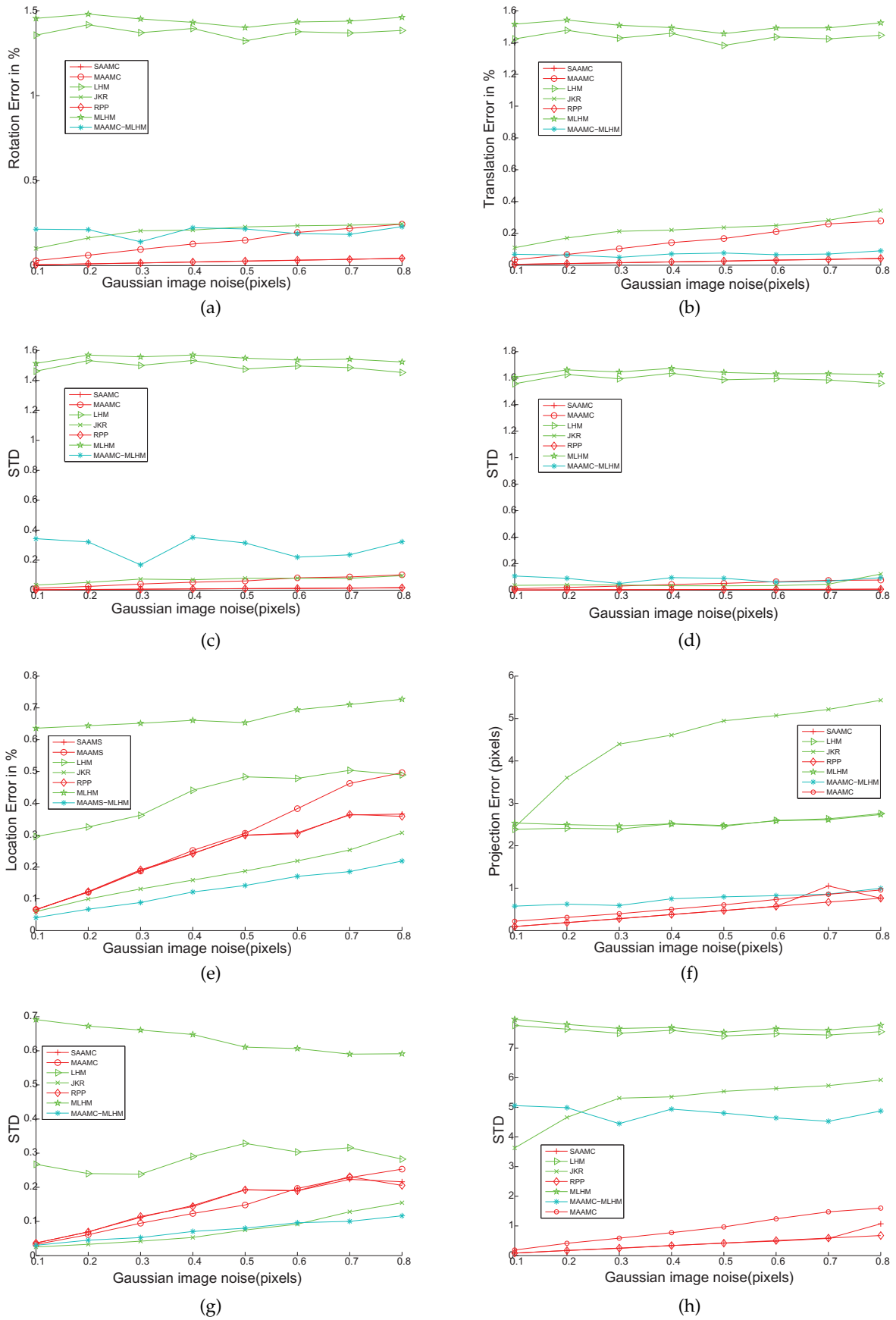


Figure 1. Simulation experiment results under varied Gaussian noise. There were two cameras in the multi-camera system using eight referenced points. (c),(d),(g) and (h) are the corresponding standard deviations of (a),(b),(e) and (f).

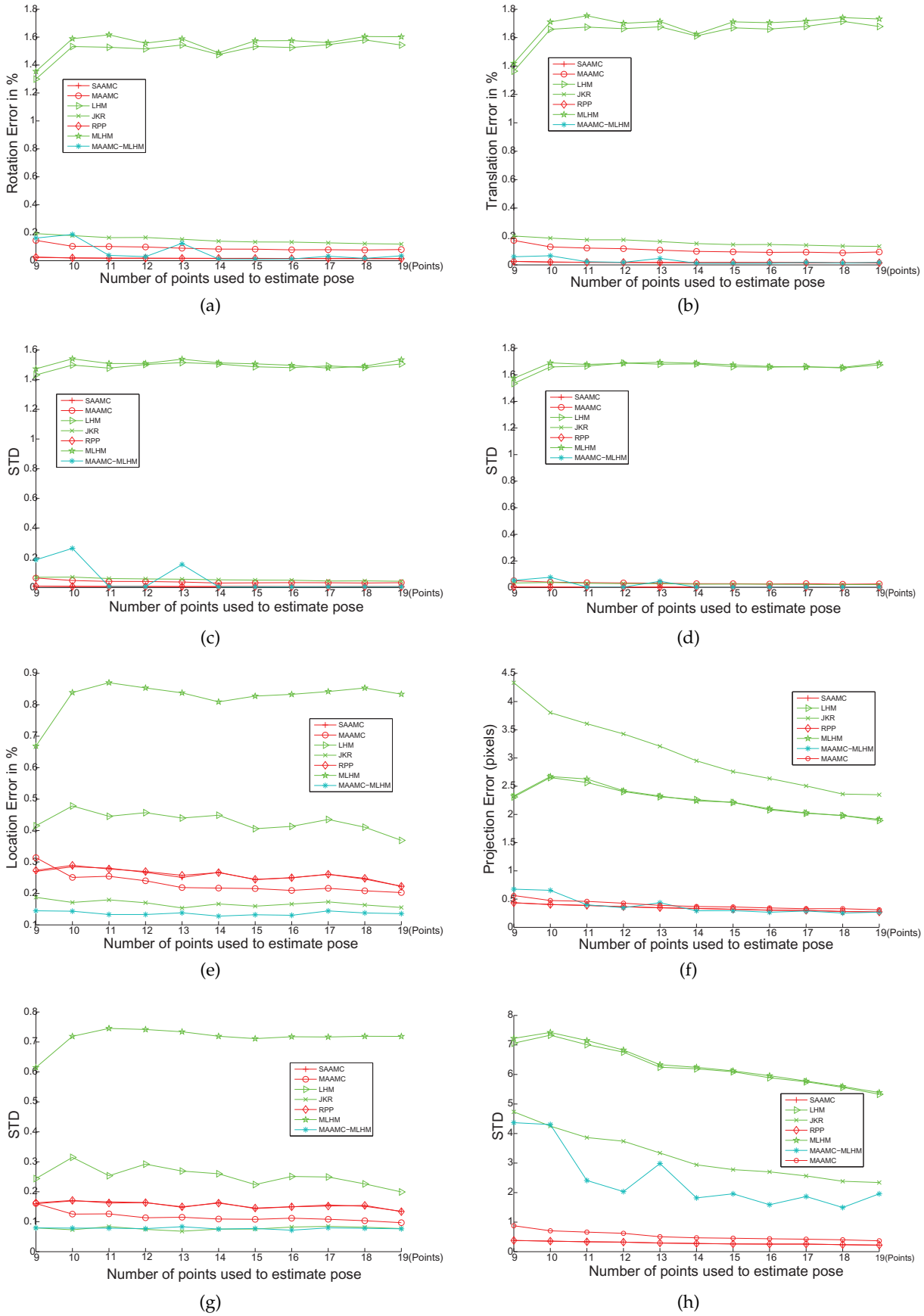


Figure 2. Simulation experiment results under varied referenced points. There were two cameras in the multi-camera system using Gaussian noise $\sigma = 0.5$. (c),(d),(g) and (h) are the corresponding standard deviations of (a),(b),(e) and (f).

sphere towards the original centre (0,0,0) with a distance of 1,500, and then executed the estimation 200 times to output the average. The results in figure 3 show that the performances of all four methods will increase as the number of cameras increases. Although the number of cameras has been increased, the $MLHM$ will also be suffered with wrong initial values - this is why the $MLHM$ performances were worst while the $MAACM - MLHM$ performances were almost the best with a good initial value from our $MAACM$.

We also compared the execution time of the four methods ($MAACM$, $MLHC$, JKR, $MAACM - MLHM$) to estimate pose simultaneously, shown in figure 4. In this experiment, we used a multi-camera system with two cameras and randomly chose the pose of the two-camera system, each method was executed 300 times and output their average. The results in figure 4 show that the computation time of JKR was the least among all four methods, as it is a linear method. Our $MAACM$ will be much faster than $MLHM$ and $MAACM - MLHM$. Although $MAACM - MLHM$ needs to execute both $MAACM$ and $MLHM$, its temporal cost is still less than $MLHM$, which indicates that the bad initial guess for $MLHM$ will lead to many more iterations and more computation.

In order to evaluate the multi-camera algorithms' robustness, we also inspected the disturbance introduced by a small calibration error of the multi-camera system. The experiment was carried out by adding Gaussian noise to the parameters of the principal point (U_0, V_0) of each camera, the relative translation vector and the rotation angles, separately. The Gaussian noise magnitudes in U_0, V_0 were 2.5 pixels, and 10 mm for all three translation orientations, and 10 degrees for all three angles correspondingly. The viewing angles were set to be large. The 3D referenced points and cameras were placed in the same manner as in the first experiment. The pose estimation target-point was also set with a ground true value (430,760,200). We executed the estimation 50 times for each noised U_0, V_0, the relative translation vector T and the relative rotation R. The average results are shown in figure 5.

According to figure 5(a) and (b), we can see that the object space-minimum-based methods (e.g., $MLHM$, $MAACM - MLHM$) will give better pose estimations than the others in relation to the rigid rig with noise, and that they will also perform fewer pose estimations than the image error-minimum-based methods (e.g., $MAACM$) as concerns the noise U_0 and V_0. In addition, figure 5(c) illustrates that our $MAACM$ will achieve the best robustness as regards the overall performance of localization.

4.1.3. Overall performance analysis of the simulation experiments

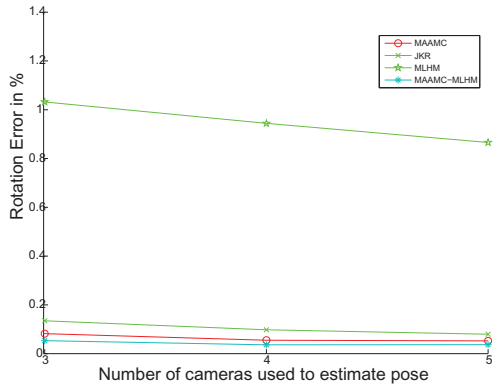
In the above simulation experiments, we have presented the performances of seven state-of-the-art pose estimation methods for various metrics. In this section, we try to present an objective analysis of our approach, in terms of both robustness and accuracy, and in comparing it with the other six methods. In our analysis, we use a comparable performance diagram to illustrate the performance of our method vs the other methods. The results are shown in figure 6. In figure 6, each row compares the performance of our $MAACM$ and the other methods for various metrics: the red block means that $MAACM$ achieves better performance with the current metrics, the green block means that $MAACM$ achieves worse performance with the current metrics, and the block with the dotted pattern means that $MAACM$ is indistinguishable from the other methods.

The results in figure 6 illustrate that $MAACM$ can achieve better performances for most of the metrics when compared to those methods obtaining poses simultaneously. As concerns the comparison between $MAACM$ and RPP, it seems that RPP may be superior to our $MAACM$. We further consider that the reason why may be because RPP could search more possible intervals of camera poses without the constraint of rigs among the cameras during the optimization. Accordingly, we carried out another comparison between RPP and $SAACM$, which is our solution for $MAACM$ when there is only one camera. The results are shown in figure 7, and they illustrate that our $SAAMC$ may be superior to RPP. Thus, our approach is better as regards comprehensive performance under varied conditions and will be appropriate for implementation in applications requiring real-time and robust pose estimation, such as augmented reality and robots, etc.

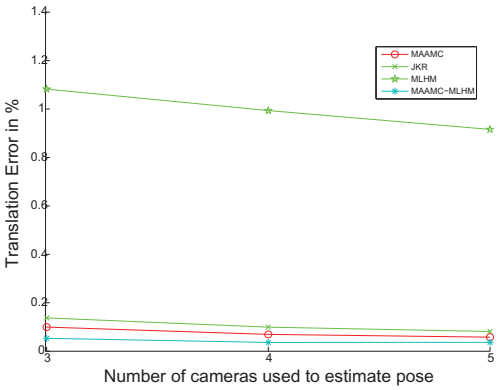
4.2 Real case experiments

We also carried out comparative experiments using real image data. In these experiments, we built a miniature multi-camera system with two Toshiba Teli cameras, as shown in figure 8(left). Each camera was equipped with a 4 mm lens and operated at a resolution of 640×480. In the experiments, there were eight green referenced points placed on the ping-pong table and one static table tennis ball as a target point, as shown in figure 8(right). We randomly placed the cameras of the system and obtained their translation poses using the methods described above. Next, we computed the positions of the table tennis ball via the pose estimated by the different methods. The true positions of the table tennis ball were obtained via a 3D Micro-hite DCC coordinate-measuring machine (CCM)[9]. There were 12 random positions for the translation pose estimation and ball location - at each position, we executed each method 200 times and removed the maximum and minimum of each method. The final output results were the average of the remaining data. The experimental results are shown in figure 9. Figure 9 shows that $MAAMC$ can achieve the highest location precision among all the methods and that it is also consistently more stable than the others.

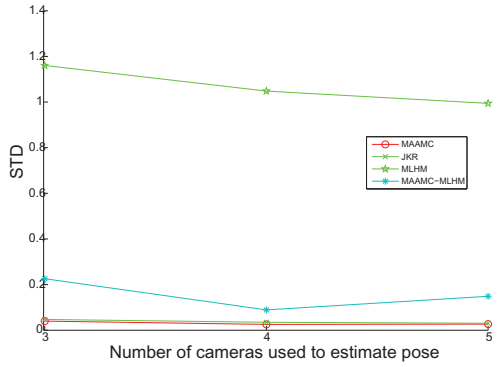
[9] For more details about the CCM, please refer to http://www.hexagonmetrology.com/ (30/04/2014)



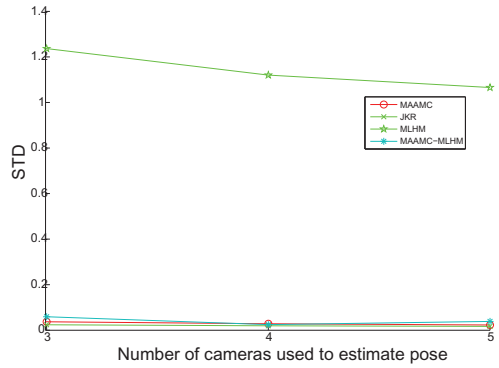
(a)



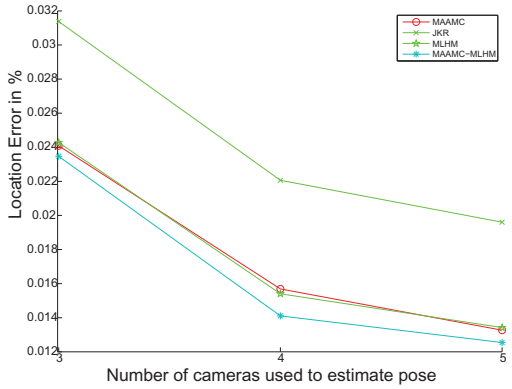
(b)



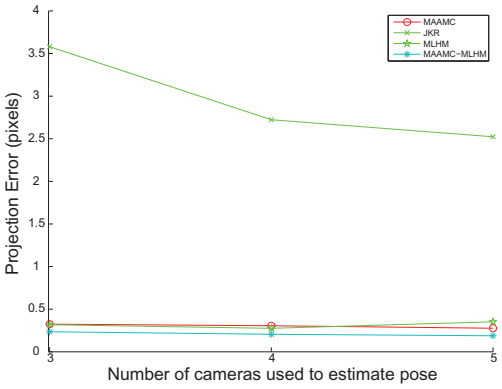
(c)



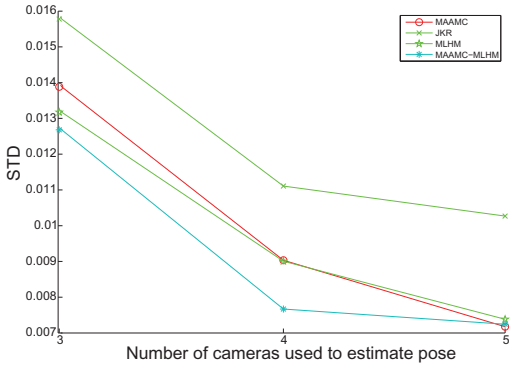
(d)



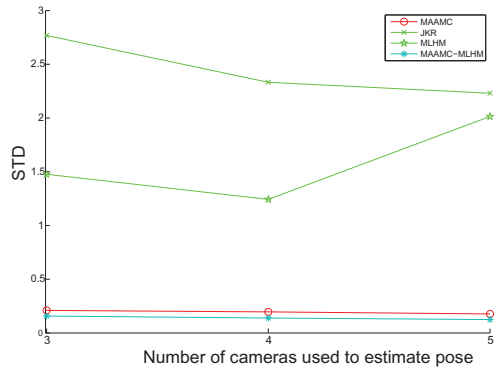
(e)



(f)

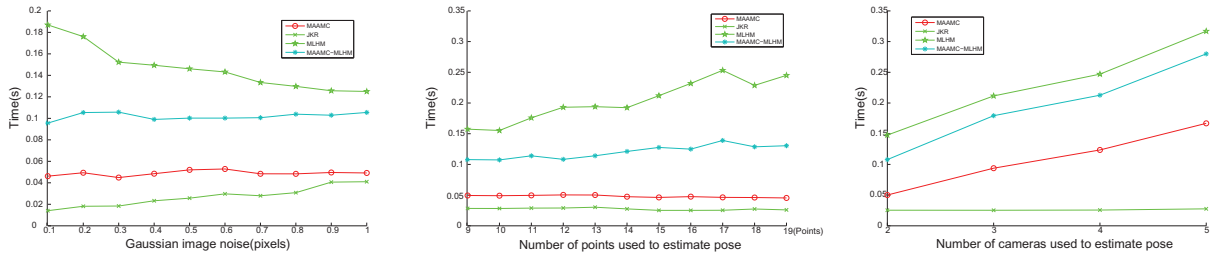


(g)



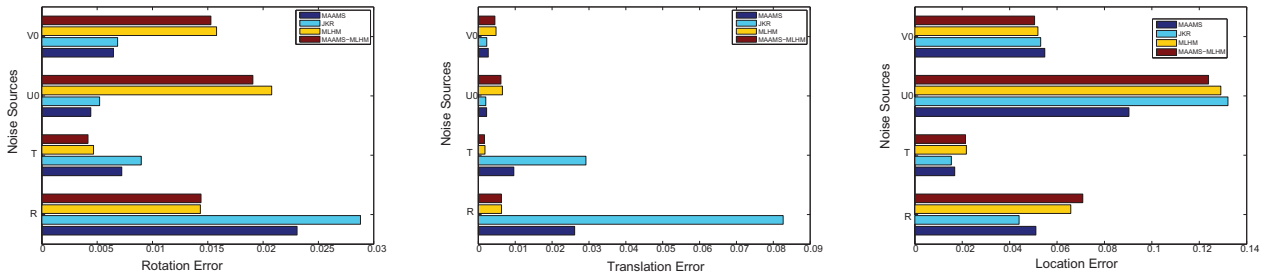
(h)

Figure 3. Simulation experiment results with increasing numbers of cameras. Gaussian noise $\sigma = 0.5$, for eight referenced points.



(a) (b) (c)

Figure 4. Experimental results for average time consumption



(a) (b) (c)

Figure 5. Simulation experiment results introducing a small calibration error. (a),(b) and (c) show the rotation, translation and location errors of the four methods with noise on their U_0, V_0 and rigid rig (R, T). The x-coordinates of these three sub-figures are relative errors, which refer to the corresponding errors compared with the absolute values, such as the values of the rotation, translation and distance.

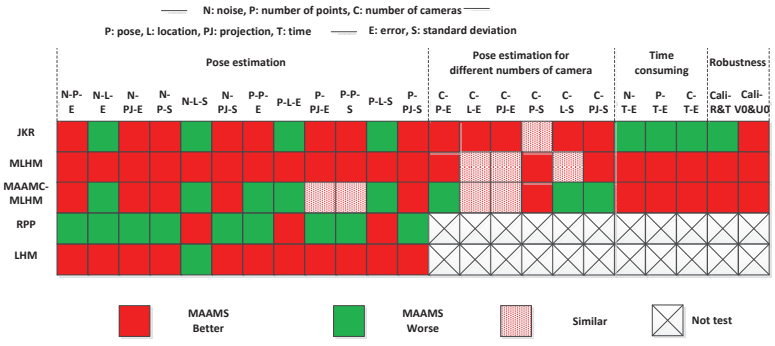


Figure 6. Overall performance comparison of $MAACM$ with other methods

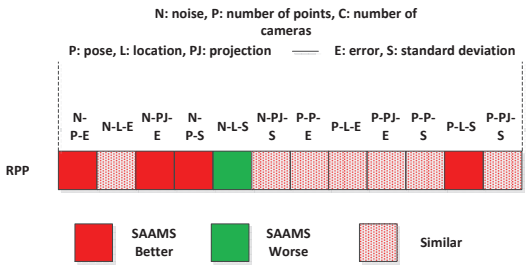


Figure 7. Overall performance comparison of $SAACM$ with RPP

Figure 8. Hardware(left) and software(right) of the multi-camera system used for real case experiments. In the right-hand figure, the upper-left view shows the 3D pose of the multi-camera system in the virtual environment, while the lower views show the images from the system and the upper-right view shows the detailed 6DOF poses.

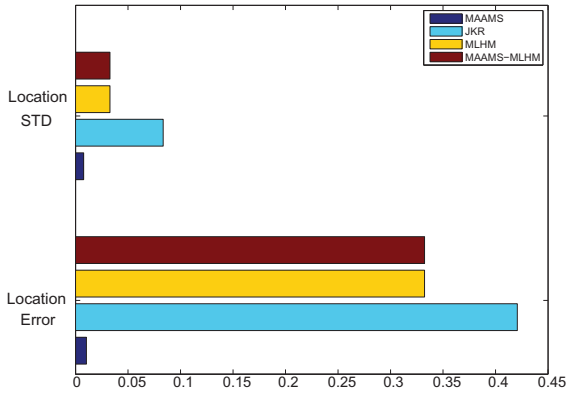


Figure 9. Localization experimental results in the ping-pong robot vision system with different methods - the x-coordinate is the ratio of the distance error compared with the absolute distance

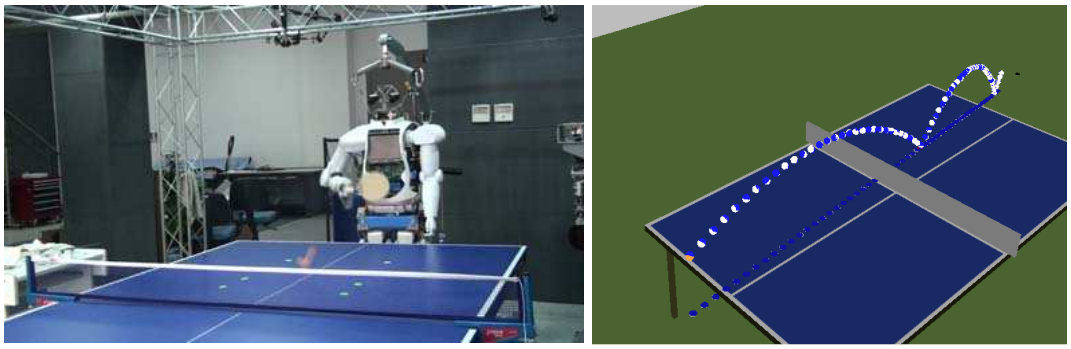


Figure 10. Left: the camera pose in our ping-pong robot system. Right: the trace obtained by our method (white) compared with the true trace (blue).

4.3 Practical implementation in ping-pong robots

The method presented in this article was originally motivated by our humanoid ping-pong robot project, which uses a multiple camera system to guide the robot arms during ping-pong games. We thus use the hardware described in Section 4.2 as the on-board vision system of the humanoid ping-pong robot. As concerns the vision system in the ping-pong robot, there are two main challenges in pose estimation due to the shaking of the robot. Firstly, there are few referenced points in this system, and the referenced 3D points are all located in the plane of the table. There are normally fewer than 10 correspondences, and so the vision system needs to estimate its pose and calculate the coordinates of the balls in the referenced point's coordinates with high accuracy using only a few coplanar points. Secondly, the vision system can be deployed at any angle relative to the ping-pong table, and so it may be placed in some locations with small viewing angles, which will greatly affect the accuracy of the pose estimation.

We implemented *MAAMC* in our humanoid ping-pong robot system [31]. In contrast to other ping-pong ball robots [32], we designed a 7DOF robot arm - which is very similar to a human arm - to play ping-pong.

The two-camera system as described in Section 4.2 tracks the ping-pong ball and generates its traces while playing. Figure 10 (right) shows the accuracy of the trace estimated by our method. The blue points show the true positions in this trace, which were obtained using an additional high-speed camera system with 500 $frames/s$ working offline, while the white points show the trace estimated by our method in real-time [10].

5. Conclusion

In this paper, we presented an efficient and robust pose estimation algorithm for multi-camera systems which can obtain 6DOF poses for all the cameras using only a few coplanar points simultaneously. Large-scale simulation experiments have shown that this algorithm can be more robust than the classical iterative pose estimation algorithm in both small- and large-angle viewing conditions. Practical experiments also showed that this method is more accurate and robust.

Hence, our method is especially suitable for implementation in tasks where there may be various poses for the cameras, including ill-condition or relatively small angles of viewing.

6. Acknowledgements

This research is based upon work supported in part by the National Science Foundation of China (61173123, 61203324, 61473258) and the Natural Science Foundation of Zhejiang (LR13F030003).

7. References

[1] Yang Yang and Qixin Cao. A fast feature points-based object tracking method for robot grasp. *International Journal of Advanced Robotic Systems*, 10(170), 2013.

[2] Sorin M. Grigorescu and Claudiu Pozna. Towards a stable robotic object manipulation through 2d-3d features tracking. *International Journal of Advanced Robotic Systems*, 10(200), 2013.

[3] H. Kato and M. Billinghurst. Developing ar applications with artoolkit. *IEEE / ACM International Symposium on Mixed and Augmented Reality*, pages 305–305, 2004.

[4] Guodong Chen, De Xu, Zaojun Fang, Zemin Jiang, and Min Tan. Visual measurement of the racket trajectory in spinning ball striking for table tennis player. *IEEE T. Instrumentation and Measurement*, 62(11):2901–2911, 2013.

[5] F. Vasconcelos, J.P. Barreto, and U. Nunes. A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2097–2107, Nov 2012.

[6] Omar Tahri, Helder Araújo, Youcef Mezouar, and François Chaumette. Efficient iterative pose estimation using an invariant to rotations. *IEEE T. Cybernetics*, 44(2):199–207, 2014.

[7] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate $o(n)$ solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.

[8] Adnan Ansar and Kostas Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, 2003.

[9] Gerald Schweighofer and Axel Pinz. Globally optimal o(n) solution to the pnp problem for general camera models. In *BMVC*, 2008.

[10] Gerald Schweighofer and Axel Pinz. Robust pose estimation from a planar target. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):2024–2030, 2006.

[11] Chien-Ping Lu, Gregory D. Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:610–622, 2000.

[12] David Nistler, Oleg Naroditsky, and James Bergen. Visual odometry. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, pages 652–659, 2004.

[13] Wen-Yan Chang and Chu-Song Chen. Pose estimation for multiple camera systems. *Pattern Recognition, International Conference on*, 3:262–265, 2004.

[14] Patrick Baker, Cornelia Fermüller, Yiannis Aloimonos, and Robert Pless. A spherical eye from multiple cameras (makes better models of the world). In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, pages 576–583, 2001.

[15] Fredrik Vikstén, Robert Söderberg, Klas Nordberg, and Christian Perwass. Increasing pose estimation performance using multi-cue integration. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, May 15-19, 2006, Orlando, Florida, USA*, pages 3760–3767, 2006.

[16] Jan-Michael Frahm, Kevin Köser, and Reinhard Koch. Pose estimation for multi-camera systems. In *DAGM-Symposium*, pages 286–293, 2004.

[17] Xu Yunxi, Jiang Yunliang, and Chen Fang. Generalized orthogonal iterative algorithm for pose estimation of multiple camera systems. *Acta Optica Sinica*, 29(1):72–71, 2009.

[18] Brian Clipp, Jae-Hak Kim, Jan-Michael Frahm, Marc Pollefeys, and Richard Hartley. Robust 6dof motion estimation for non-overlapping, multi-camera systems. In *WACV '08: Proceedings of the 2008 IEEE Workshop on Applications of Computer Vision*, pages 1–8, Washington, DC, USA, 2008. IEEE Computer Society.

[19] Jae-Hak Kim, Richard I. Hartley, Jan-Michael Frahm, and Marc Pollefeys. Visual odometry for non-overlapping views using second-order cone programming. In *omputer Vision - ACCV 2007, 8th Asian Conference on Computer Vision, Tokyo, Japan, November 18-22, 2007, Proceedings, Part II*, pages 353–362, 2007.

[10] For more results for our ping-pong robot, please refer to: http://www.youtube.com/watch? v=t_qN3dgYGqE (30/04/2014)

[20] P. Azad, T. Asfour, and R. Dillmann. Stereo-based 6d object localization for grasping with humanoid robot systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 919–924, 2007.

[21] Jae-Hak Kim, Hongdong Li, and Richard I. Hartley. Motion estimation for multi-camera systems using global optimization. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008.

[22] Denis Oberkampf, Daniel F. DeMenthon, and Larry S. Davis. Iterative pose estimation using coplanar feature points. *Computer Vision and Image Understanding*, 63(3):495–511, 1996.

[23] Toshio Ueshiba and Fumiaki Tomita. Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices. In *ICCV*, pages 966–973, 2003.

[24] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 2000.

[25] Ce Liu, Richard Szeliski, Sing Bing Kang, C. Lawrence Zitnick, and William T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):299–314, 2008.

[26] Siwei Lyu and Eero P. Simoncelli. Statistical modeling of images with fields of gaussian scale mixtures. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 945–952, 2006.

[27] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics.*, 22(3):313–318, 2003.

[28] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[29] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.

[30] M. Ahmed and Aly A. Farag. Nonmetric calibration of camera lens distortion: differential methods and robust estimation. *IEEE Transactions on Image Processing*, 14(8):1215–1230, 2005.

[31] Rong Xiong, Liu Yong, and Hongbo Zheng. A humanoid robot for table tennis playing. In *ARSO*, pages 66–67, 2011.

[32] Zhengtao Zhang, De Xu, and Junzhi Yu. Research and latest development of ping-pong robot player. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 4881–4886, June 2008.