Simultaneous Hand-Eye Calibration and Reconstruction

Xiangyang Zhi and Sören Schwertfeger[1]

***Abstract*— Hand-eye calibration is a well-known calibration problem. The problem assumes that a camera (eye) is rigidly mounted to the gripper (hand) of a robot arm and aims to find the transformation between them. In this paper, we propose a novel pipeline for hand-eye calibration without the use of a calibration target.**

First we employ feature extraction and matching, followed by an initial hand-eye calibration step using 2-view matches. In an iterative process, we then alternately employ triangulation and bundle adjustment to optimize the reconstruction and the hand-eye calibration result. Unlike in structure from motion and traditional hand-eye calibration, during this process we always determine the global camera poses using the hand poses and the estimated hand-eye transformation.

Synthetic-data and real-data experiments are performed to evaluate the proposed approach, and the results indicate that the accuracy of our approach is superior to state-of-the-art approaches. Moreover, the speed of our algorithm is faster than existing methods.

I. Introduction

Calibration is a common problem in computer vision and robotics. Let's consider a special calibration case: suppose that we have a robot arm with a gripper as its end-effector. In manipulation scenarios, to help locate the position of the target, a camera is mounted rigidly on the end-effector. If we use an object recognition algorithm to obtain the pose of the target in the frame of reference of the camera, we need to know the transform between the camera (eye) and gripper (hand), in order to calculate the pose of the target with respect to the gripper. This problem has been studied abundantly in past years and is well known as hand-eye calibration.

Unlike camera-camera calibration in stereo vision, one cannot expect to extract feature correspondences between the gripper and the camera. However, this doesn't mean that hand-eye calibration is impossible. On the contrary, since no feature correspondence are needed we can apply the solutions of the hand-eye calibration problem to various other calibration applications, *e.g.* camera-laser, camera-IMU, laser-IMU and so on.

Hand-eye calibration is based on motion. Suppose that we have a series of corresponding hand and eye motions, and A_{ij} and B_{ij} are the homogeneous transformations of the eye and hand motions, respectively. Therefore, the equation

$$A_{ij}X = XB_{ij} \tag{1}$$

is an obvious observation from Figure 1, where X is the unknown transformation from gripper to camera. It is noteworthy that some publications also use another kind of expression: they describe X as the gripper pose with respect to the camera frame, which is the same as our description. X is the same at all times since the camera is rigidly mounted on the arm. Tsai *et al.* [1] proved that two or more motions with nonparallel rotation axes can determine X uniquely. We assume a fully calibrated robot arm, such that the pose of the gripper relative to robot arm base, denoted by B_i in Figure 1, can be calculated using the measured joint angles and the forward kinematic model of the arm. We can thus obtain relative hand motions from the i^{th} hand frame to the j^{th} hand frame:

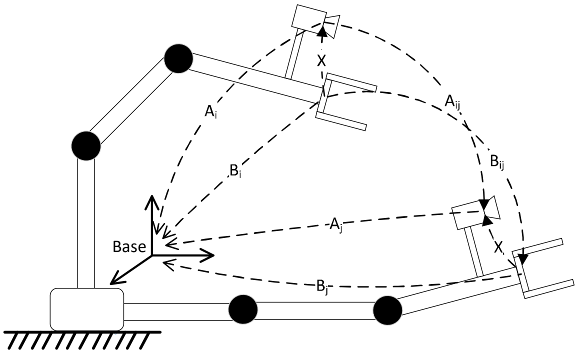

Fig. 1. Hand-eye calibration: X is the unknown transformation from gripper to camera (the gripper pose with respect to the camera frame), A_{ij} is the transformation between the camera frame at times i and j, so representing the camera motion. B_{ij} is the corresponding transformation of the motion of the end-effector between times i and j. The following relation can be easily observed: $A_{ij}X = XB_{ij}$. B_i and B_j, and thus B_{ij}, can be obtained using the forward kinematics of the robot arm, while A_i and A_j are not known.

$$B_{ij} = B_j^{-1}B_i \tag{2}$$

The camera motions are usually determined using structure from motion (SFM) algorithms [2]. However, if there is no target with known geometry, the calculated translation will be up to an unknown scale factor. We split Eq. (1) into the rotation and translation parts:

$$R_{A_{ij}}R_X = R_X R_{B_{ij}} \tag{3}$$

$$R_{A_{ij}}t_X + \lambda_{ij}t_{A_{ij}} = R_X t_{B_{ij}} + t_X \tag{4}$$

where R denotes the rotation and t the translation part of a transform, *e.g.* $A_{ij} = \{R_{A_{ij}}, t_{A_{ij}}\}$. λ_{ij} is the unknown scale factor. In all present approaches which utilize Eq. (1) or (3) and (4) to solve X, the unknown scale factor is regarded to be constant for all motions. In particular, when there is a known target, like a chess pattern [3], the camera and hand motion can be transformed to the same metric, thus λ_{ij} is 1 in this situation.

[1] Both authors are with the School of Information Science and Technology, ShanghaiTech University, 200031 Shanghai, China, [zhixy, soerensch]@shanghaitech.edu.cn

This paper presents a novel pipeline for hand-eye calibration. In the first step we extract features from all images and match them. Secondly, according to feature matches between two views, we estimate the relative poses of 2-view matches. Thirdly, with hand poses and relative camera poses of 2-view matches, we form several equations of (3) and (4) and solve the hand-eye transformation. However, this initial calibration is extremely inaccurate, so we employ triangulation and bundle adjustment alternately for several times, in order to refine hand-eye calibration and 3D reconstruction simultaneously.

The contributions of this paper are three-fold. (I) we propose a novel end-to-end pipeline for hand-eye calibration. Our approach utilizes known tools and techniques, but uses them in a unique way: We combine 3D reconstruction and hand-eye calibration. Thus our algorithm differs significantly from existing methods which usually use SFM just as an input source to get the relative camera motions. (II) we extend $AX = XB$ by supposing that any two camera motions are up to an unequal scale factor, and propose a new method to solve this problem. (III) we demonstrate the high accuracy of our approach with synthetic and real data experiments.

The remainder of this paper is arranged as follows. Section II discusses related work. Section III gives an overview of our algorithm. Section IV introduces the initial hand-eye calibration algorithm by solving $AX = XB$ and Section V explains the bundle adjustment. Section VI presents experiments to analyze and compare the performance of our algorithm with others. Section VII gives a conclusion of this work.

II. Related Work

The hand-eye calibration problem was proposed in the 1980s and has been studied for nearly 40 years. There is a considerable number of publications about hand-eye calibration so far and new ones are still appearing continuously. On the other side, the hand-eye calibration problem itself is developing and has expanded to several branches. In the rest of this section, we will review a few hand-eye calibration algorithms by branches.

A. Conventional Hand-Eye Calibration

At the very beginning, hand-eye calibration was regarded as to solve Eq. (1), and hand and camera motion are both with real metric, *i.e.* λ_{ij} in Eq. (4) is 1. We call this kind of problem *conventional hand-eye calibration*. Conventional hand-eye calibration was presented earliest, so there is lots of literature on that topic.

Early approaches [1], [4]–[6] treated this problem as two separate parts: rotation and translation. They firstly estimated R_X using Eq. (3), then utilized the estimated R_X to solve t_X using Eq. (4). However, such a separation inevitably leads to the propagation of the residual error of the estimated rotation into the translation estimation, so later approaches for simultaneous estimation of both R_X and t_X appeared. Daniilidis [7] derived a linear formulation using the dual quaternion representation of Eq. (3) and (4), and then SVD was utilized to solve R_X and t_X simultaneously. As a part of [8], Andreff *et al.* also proposed a linear formulation of Eq. (1) using the Kronecker product.

Several iterative methods were proposed in past years. Zhuang and Shiu [9] presented an iterative nonlinear method to minimize the loss function $\sum_{i,j} \|A_{ij}X - XB_{ij}\|^2$ to estimate X. In [10], Horaud and Dornaika separated the rotation and translation parts, and used nonlinear optimization to minimize the sum of the norm of the rotation and translation parts. The disadvantage of the nonlinear method is that an initial solution is necessary, otherwise, it may not converge to the global optima. In [11], Zhao utilized the linear formulation proposed in [8] and [7] to construct the so-called *second-order cone program* (SOCP) problem [12]. SOCP is a kind of convex problem where it is guaranteed to find the global optima even without an initial solution. However, Zhao's approach doesn't enforce the orthogonality of the rotation matrix, such that an orthogonalization progress is necessary which will increase the error of the rotation estimation.

B. Extended Hand-Eye Calibration

Now, let's consider that λ_{ij} in Eq. (4) is unknown. In this case, conventional hand-eye calibration approaches don't work any more, but since the hand's poses are in real metric units, the rigid transformation can still be solved. This problem has no unified name, some authors call it online hand-eye calibration or SFM based hand-eye calibration. We prefer the name *extended hand-eye calibration*, for this problem is a more general case which includes the conventional hand-eye calibration. Extended hand-eye calibration approaches can be used to solve the conventional hand-eye calibration problem directly. We will classify any approach that can be used to solve the extended hand-eye calibration as extended hand-eye calibration approach, even if the authors didn't announce that explicitly. The method proposed in this paper also belongs to extended hand-eye calibration approaches.

The first solution to extended hand-eye calibration was proposed by Andreff *et al.* [8]. A linear formulation of the hand-eye calibration was derived using the Kronecker product, and Andreff *et al.* suggested to estimate R_X and t_X separately. Schmidt *et al.* [13] exploited two kinds of non-linear optimization techniques using Eq. (3) and (4) and their dual quaternion representation, respectively. Heller *et al.* [14] firstly estimated R_X using the method proposed by Park *et al.* [6], and then estimated t_X using SOCP.

It is noteworthy that camera motions estimated by SFM are usually noisy, for there often exist wrong feature correspondences, especially in environments with repetitive structures. Such noise will increase the error in the camera pose estimation. As a result, the noisy A_{ij} will mislead the extended hand-eye calibration using $AX = XB$. In [15] and [16], Heller *et al.* and Ruland *et al.* simultaneously proposed very similar branch-and-bound search methods, which can ensure the global optimum with respect to the L_∞-norm. A slightly extended approach of [15] is given in [17]. We notice that the branch-and-bound search approach utilizes feature correspondences rather than camera motions, thus overcoming the disadvantage of traditional extended hand-eye calibration, which makes it the state-of-the-art method to our best knowledge. However, this branch-and-bound search method is more than ten times slower than traditional approaches.

C. Simultaneous Hand-Eye and Robot-World Calibration

Another branch of hand-eye calibration is *simultaneous hand-eye and robot-world calibration*. This problem was first proposed by Zhuang *et al.* [18]. The additional robot-world calibration is to determine the transformation between the robot base frame and world frame which is usually based on a chess pattern. For that reason, to our best knowledge, all approaches for simultaneous hand-eye and robot-world calibration assume that the camera poses are in real metric units.

Similar to conventional hand-eye calibration, approaches for simultaneous hand-eye and robot-world calibration can also be classified by estimating rotation and translation separately or simultaneously. Early approaches usually estimated the rotation and translation separately [12], [18]–[20]. As a part of [12], Dornaika and Horaud proposed a nonlinear approach which estimates rotation and translation simultaneously. Li *et al.* [21] utilized the Kronecker product and dual quaternions to do so. In [22], Malti proposed a bundle adjustment based approach which is similar to the bundle adjustment partition of our algorithm. However, the difference is that Malti's approach doesn't optimize the position of 3D points while our approach does.

Recently, Wu *et al.* [23] extended hand-eye calibration to a even more complicated case, called *hand-eye, tool-flange and robot-robot calibration*, which could be used in multi-robot cooperation.

One common fact for all three branches of hand-eye calibration is that they need hand and eye motion correspondences, but due to the asynchronicity of different sensors, the correspondences may not hold. As a result, several probabilistic approaches [24]–[28] are proposed to solve this kind of problem. Due to less correlation to this work, we won't explain them here.

III. Overview

For robotic arms the pose of the gripper depends on the angles of all arm joints, which are determined by high accuracy encoders. As a result, the gripper pose can be calculated with high precision (the repeat localization precision is usually better than 0.2mm). However, the accuracy of SFM highly depends on the performance of the feature matching. Bad feature correspondences will result in large errors in the camera pose estimation. Unfortunately, there is no absolutely perfect feature matching so far. To make things worse, the intrinsic parameters of a camera are also estimated by SFM, and inaccurate intrinsic parameters will reduce the pose estimation accuracy. For all these reasons, the traditional hand-eye calibration algorithms, which are based on solving $AX = XB$, are restricted by the camera pose estimation. In our algorithm, we only utilize $AX = XB$ to guess the hand-eye transformation initially, but then bundle adjustment is utilized to obtain a more accurate calibration result. In other words, we avoid relying on the estimated SFM camera poses. The experiments in Section VI will illustrate that the accuracy of our hand-eye calibration is significantly improved.

Figure 2 shows a diagram of our algorithm. At the beginning feature extraction and matching are executed. This is a common step in computer vision, so the details are omitted in this paper. After matching features, the relative poses of two views which have enough correspondences can be determined. If the readers would like to learn the basis of 2-view geometry, we refer to [2]. We assume the cameras to be already intrinsically calibrated. With two or more pairs of hand and camera motions, we can estimate the rigid transformation between the hand and the camera by solving $AX = XB$, which will be explained in Section IV. This serves as the initial calibration value for the next steps.

Suppose that we know the transformation from hand to camera (X) and one of the hand poses, say B_i, then we can calculate the corresponding camera pose with respect to robot arm base, say A_i (see Figure 1), using:

$$A_i = B_i X^{-1} \tag{5}$$

As we have stated before, the hand poses are usually very accurate, so if the hand-eye transformation is estimated precisely, so are the eye poses calculated by Eq. (5). Therefore, instead of estimating camera pose by SFM, we utilize Eq. (5) in triangulation and bundle adjustment.

Triangulation is a fundamental problem in 3D reconstruction. Given a 3D point's projections onto two or more images, triangulation aims to estimate the 3D position of the point. Its geometrical principle can be found in [2]. In triangulation, the camera poses are needed, so that Eq. (5) will be utilized in our algorithm. It should be noted that since the camera pose determined by Eq. (5) is in real metric units, the positions of the points will also be in real metric units.

As the most important step of our algorithm we employ bundle adjustment [29] to refine the hand-eye calibration and the reconstruction simultaneously. Details about bundle adjustment will be given in Section V. At the end, our approach will output the final hand-eye calibration result, and the reconstruction as the by-product.

Please notice an additional advantage of our approach. If we denote a view by a vertex and two vertices are connected by an edge if two views are matched (their relative transform can be calculated based on matched features), we get a view graph. SFM based reconstruction can only be done on a connected subgraph. As a result, traditional hand-eye calibration can also only work on connected subgraphs independently, as they require the camera positions to be within the same metric. However, this limitation doesn't appear in our approach, as our method does not estimate the global camera pose using SFM. Moreover, the several reconstructions estimated with subgraphs can even be combined into one. This ability of our algorithm is very useful for robot arms whose motion space is really restricted.

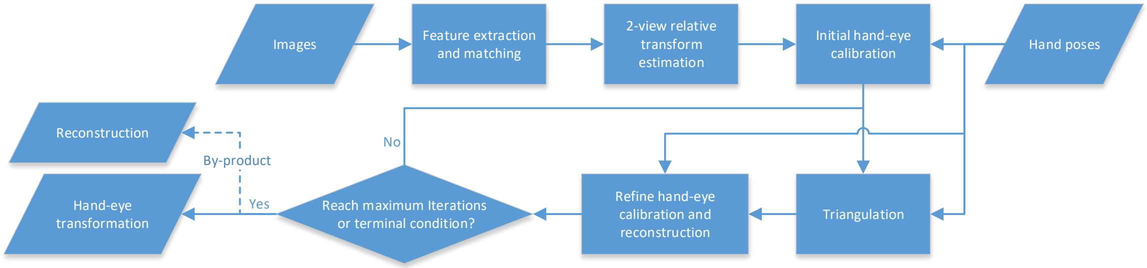
Images
Feature extraction and matching
2-view relative transform estimation
Initial hand-eye calibration
Hand poses
Reconstruction
By-product
No
Yes
Hand-eye transformation
Reach maximum Iterations or terminal condition?
Refine hand-eye calibration and reconstruction
Triangulation

Fig. 2. Block diagram of the proposed hand-eye calibration algorithm.

The C++ code of our algorithm is available on `https://github.com/STAR-Center`.

IV. Initial Hand-Eye Calibration

As we have stated in Section II, all existing extended hand-eye calibration algorithms, which utilize Eq. (1) or (3) and (4) to solve X, assume that λ_{ij} is unknown but constant for all hand-eye motions, because they have estimated the global camera poses before forming $AX = XB$. However, in this work, we attempt to utilize the transformations estimated with 2-view matches directly. Since there is no known object, we notice that λ_{ij} can be variable for any two different translations of 2-view matches, *i.e.*

$$\lambda_{ij} \neq \lambda_{mn},\ if\ i \neq m\ or\ j \neq n \tag{6}$$

As a result, we actually propose an extension for the extended hand-eye calibration problem, and obviously, traditional algorithms for extended hand-eye calibration are unavailable to solve it. To solve this new problem we propose a novel algorithm based on eliminating λ_{ij}. Before that, let's introduce a definition.

Definition 1 (Skew-symmetric matrix). Suppose that $u = [u_1, u_2, u_3]^T \in \mathbb{R}^3$, we define its corresponding skew-symmetric matrix, denoted by $\hat{u}$, as:

$$\hat{u} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3} \tag{7}$$

Obviously, $\hat{u}$ is a 3×3 skew-symmetric matrix, *i.e.* $\hat{u}^T = -\hat{u}$. It can be immediately verified that $\hat{u}u = 0$. For Eq. (4), we multiply both sides by $\hat{t}_{A_{ij}}$, obtaining

$$\hat{t}_{A_{ij}} R_{A_{ij}} t_X = \hat{t}_{A_{ij}} R_X t_{B_{ij}} + \hat{t}_{A_{ij}} t_X \tag{8}$$

Now λ_{ij} is eliminated and traditional methods to estimate t_X with known R_X, *e.g.* [1], [5], [6], [8], can be used to solve the extended hand-eye calibration. In this work we propose a SVD-based approach, which is inspired by Andreff *et al.* [8].

Neudecker [30] proved that the following equation is true:

$$vec(XYZ) = (X \otimes Z^T) vec(Y) \tag{9}$$

where X, Y and Z are any dimension-compatible matrices, $\otimes$ denotes the Kronecker product and *vec* is an operator which reorders the entries of a *m* by *n* matrix, say V, to a column vector, *i.e.* row-wise vectorization:

$$vec(V) = [v_{11}, v_{12}, \cdots, v_{1n}, v_{21}, v_{22}, \cdots, v_{mn}]^T \tag{10}$$

vec^{-1} is defined as the inverse operator to vec, *i.e.* $vec^{-1}(vec(V)) = V$.

According to Eq. (9), we can rewrite and concatenate Eq. (3) and (8):

$$\begin{bmatrix} I_9 - R_{A_{ij}} \otimes R_{B_{ij}} & 0_{9\times3} \\ \hat{t}_{A_{ij}} \otimes t_{B_{ij}}^T & \hat{t}_{A_{ij}}(I_3 - R_{A_{ij}}) \end{bmatrix} \begin{bmatrix} vec(R_X) \\ t_X \end{bmatrix} = \begin{bmatrix} 0_{9\times1} \\ 0_{3\times1} \end{bmatrix} \tag{11}$$

Suppose there are n pairs of hand-eye motions, then $12n$ linear equations can be determined, such that the coefficient matrix, say C, is 12n by 12. The solution of Eq. (11) is in a 1-dimensional subspace, which is just the null space of C, so that we can utilize SVD to find it. Yet, there is another constraint that R_X has to be a rotation matrix, *i.e.* the determinant of R_X is 1. According to this property of R_X, we can determine the unique solution.

Suppose that the solution of Eq. (11) is x, we denote $M = vec^{-1}(x_{1:9})$, where $x_{1:9}$ are the first nine entries of x. If without noise, M and R_X satisfy

$$M = \alpha R_X \tag{12}$$

where $\alpha \in \mathbb{R}$, such that

$$\alpha = (det(M))^{\frac{1}{3}} \tag{13}$$

With α, the estimated hand-eye rotation and translation are

$$\tilde{R}_X = \frac{M}{\alpha} \tag{14}$$

$$\tilde{t}_X = \frac{x_{10:12}}{\alpha} \tag{15}$$

Due to noise, $\tilde{R}_X$ is usually not orthogonal, so one more orthogonality procedure is necessary. We are employing QR decomposition for that.

A. Selection of Hand-Eye Motions using RANSAC

2-view matches are rough, so there exists a significant number of outliers in the estimated transformations of 2-views matches, and these outliers will inevitably reduce the accuracy of the initial hand-eye calibration. RANSAC [31] is a simple but useful algorithm to reject outliers, which is commonly used in computer vision. In this work, we utilize it to improve the accuracy of the initial hand-eye calibration.

One RANSAC sample is generated by randomly selecting a certain number of 2-view matches and solving the inital hand-eye calibration from Section IV for them. As two pairs of hand-eye motions could determine the unique hand-eye transform, in our experiments we used two pairs of 2-view matches. Then we calculate the error e for all valid 2-views and thus determine the number of inliers for this sample. We rely on a good estimate of the error in order to determine if a sample is an inlier or outlier. The first step in the computation of the error e is that we predict A_{ij} using Eq. (3) and (4):

$$\tilde{R}_{A_{ij}} = R_X R_{B_{ij}} R_X^T \tag{16}$$

$$\tilde{t}_{A_{ij}} = R_X t_{B_{ij}} + (I_3 - \tilde{R}_{A_{ij}}) t_X \tag{17}$$

The rotation error e_R is defined as follows:

$$e_R = \|R_{A_{ij}} - \tilde{R}_{A_{ij}}\|_2 \tag{18}$$

Although λ_{ij} is not estimated, we assume that λ_{ij} is the one enabling $\tilde{t}_{A_{ij}}$ to approach $t_{A_{ij}}$ as close as possible, so we define the translation error as follows:

$$e_t = \min_{\lambda_{ij}} \|\lambda_{ij} t_{A_{ij}} - \tilde{t}_{A_{ij}}\|_2 \tag{19}$$

This is a quadratic function about λ_{ij} and one can prove that its minimum is:

$$e_t = \sqrt{\|\tilde{t}_{A_{ij}}\|_2^2 - \langle t_{A_{ij}}, \tilde{t}_{A_{ij}}\rangle^2 / \|t_{A_{ij}}\|_2^2} \tag{20}$$

The final error e is a combination of the rotation and translation errors. In our experiments, we always set the translation unit to meter, so in order to balance the rotation and translation errors, we scale the translation error by 0.1, *i.e.* the total error is

$$e = e_R + 0.1 * e_t \tag{21}$$

In this work, we let the threshold of the error be 0.01, the minimal inlier ratio be 60%. It should be noted that this is just an initial guess, it is unworthy to spend much time to get a small accuracy improvement, so we set the maximal iterations of RANSAC to 500.

V. Bundle Adjustment

Bundle adjustment almost always appears in image-based 3D reconstruction algorithms, refining camera parameters and positions of 3D points. Specifically speaking, it aims to minimize the overall reprojection error of each estimated point to the image which can observe that point. In mathematical expression: assume that there are *m* 3D points and *n* views. x_{ij} denotes the projection of the i^{th} point in the j^{th} view, and v_{ij} is equal to 1 if the i^{th} point can be observed by the j^{th} view, otherwise 0. Moreover, assume A_j is the rigid homogeneous transformation from the j^{th} view frame to the world frame, $P_j(\cdot)$ is the projection of the j^{th} view including distortion, and let T_i be the predicted homogeneous position of the i^{th} point. The bundle adjustment model is formed as:

$$\min_{P_j, A_j, T_i} \sum_{i=1}^{m} \sum_{j=1}^{n} v_{ij} \|x_{ij} - P_j(A_j^{-1} T_i)\|_2^2 \tag{22}$$

In our algorithm, we utilize Eq. (5) to substitute A_j for bundle adjustment, so we will refine the hand-eye transformation and keep the hand poses constant. Besides, we assume that the camera has been calibrated, so that we won't refine camera intrinsic parameters, although we actually could. Our resulting bundle adjustment model is:

$$\min_{X, T_i} \sum_{i=1}^{m} \sum_{j=1}^{n} v_{ij} \|x_{ij} - P_j(X B_j^{-1} T_i)\|_2^2 \tag{23}$$

The hand-eye transformation consists of 3 rotation parameters and 3 translation parameters, while each point consists of 3 position parameters, *i.e.* there are $3m + 6$ variables in (23). Substantially similar to (22), (23) can also be solved using the Levenberg-Marquardt (LM) approach.

The initial guess may be very inaccurate, so as to improve that, we perform the triangulation and bundle adjustment alternately for several times. Considering that the hand-eye translation is in real metric units, we can set the precision we would like to get. In this work, the iteration will terminate when estimated hand-eye translation changes less than $10^{-6} m$ compared to that of last iteration or reaches the maximum number of iterations, which is 10 in this paper.

VI. Experiments

We verify our algorithm with various different experiments. We implement our algorithm based on Theia [32], an end-to-end C++ SFM library. We utilize SIFT [33] as the feature descriptor and the Ceres solver [34] as bundle adjustment solver. In real-data experiments, we utilize Bouguet's camera calibration toolbox [35] to obtain the intrinsic and extrinsic parameters of the camera.

We compare our algorithm with three other approaches. To be convenient, in the following experiments, we will denote our algorithm as "Ours". Besides, we denote the approach proposed by Andreff *et al.* in [8] as "Andreff". Since "Andreff" needs camera motions, an incremental SFM pipeline implemented by Theia is performed in advance. We denote the approach proposed by Heller *et al.* in [14], [15] as "Heller11" and "Heller12", respectively, and feature extraction and matching for both of them comes from Theia, too.

"Ours", "Andreff" and "Heller12" are all implemented with C++, and we conduct them on the same 64-bit Linux platform with an Intel Core i7-4790 3.60GHz processor. "Heller11" is implemented with MATLAB, without using multi threading. Therefore, the readers will notice that, in the following experiments, the runtime of "Heller11" is much longer than the other algorithms.

A. Synthetic-data Experiment

We first evaluate our approach with a synthetic scene and a virtual camera. 500 points are generated randomly inside a cuboid. The center of the cuboid locates at $(5, 0, 0)$, and the edge lengths along x, y and z axis are 2m, 5m and 5m, respectively. The intrinsic parameters of the virtual camera are listed in Table I. We will compare the performance of

TABLE I

INTRINSIC PARAMETERS OF THE VIRTUAL CAMERA FOR THE SYNTHETIC EXPERIMENT

Parameters	Value	Parameters	Value
Image width	3000	aspect ratio	1
Image height	2000	skew	0
Principle point	(1500,1000)	distortion	None
Focal length	2200		

TABLE II

RUNTIME COMPARISON FOR THE SYNTHETIC-DATA EXPERIMENT.

Approach	Andreff	Heller11	Heller12[1]	Ours
Runtime	0.38s	1800s	90s	0.19s

different algorithms regarding two factors: projection noise and number of hand-eye motions.

Projection Noise 15 camera positions are generated inside a cube whose center is the coordinate origin and edge length is 3m. With its position, the camera's orientation is determined by assuming that the camera looks towards the 3D point cuboid center, *i.e.* $(5, 0, 0)$. Every point will be projected to the image plane at every view, but the projection will be neglected if it is outside the image. Moreover, every projection will be corrupted by Gaussian noise in the pixel domain with a standard deviation $\sigma \in [0, 3.0]$ and a step of 0.5 pixel. Finally, we randomly select a hand-eye transformation, with a random rotation and a translation of up to 0.2m along each axis, and the hand poses are calculated using $B_i = A_i X$.

Number of Hand-Eye Motions In this experiment, we fix the σ to be 0.2, and the hand-eye motion varies from 6 to 12 with a step of 1, while keeping all other parameters same to "projection noise" experiment.

To qualify the results, we compute the errors associated with rotation and translation as follows:

$$e_R = \|\tilde{R}_X - R_X\|_2 \tag{24}$$

$$e_t = \frac{\|\tilde{t}_X - t_X\|_2}{\|t_X\|_2} \tag{25}$$

where $\tilde{R}_X$ is the estimated rotation, R_X is the true rotation, $\tilde{t}_X$ is the estimated translation and t_X is the true translation. For each noise level, 30 experiments were done independently, and the final error is the mean of all 30 errors.

Using a box plot, Figure 3 illustrates the error distributions of the hand-eye rotation and translation estimated by the four algorithms. Clearly, our approach beats the others both in rotation and translation estimation. Meanwhile, "Heller12" performs worst. The error of rotation estimated by "Heller11" is smaller than that estimated by "Andreff". However, "Andreff" performs better regarding translation estimation. We believe that the reason is that the residual error of the estimated rotation propagates into the translation estimation, because "Heler11" estimates rotation first, while the translation is calculated afterwards using the estimated rotation.

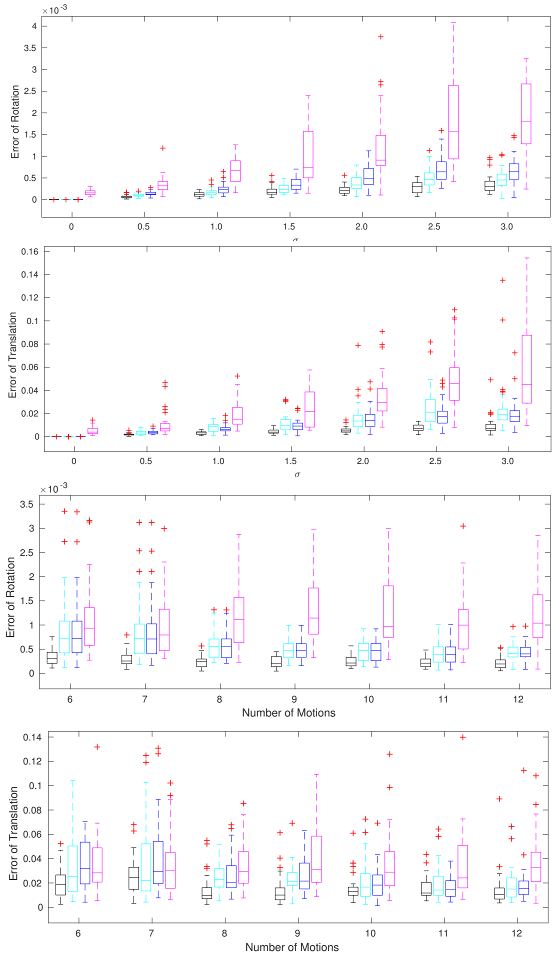

Fig. 3. Error of estimated rotation and translation against σ (above two figures) or number of motions (below two figures). For each standard deviation the boxes show the error of the following algorithms: "Ours" (left-most, black), "Heller11" (cyan), "Andreff" (blue) and "Heller12" (rightmost, magenta). To save space, some large outliers which are denoted by '+' in red are not shown.

Table II gives the average runtime of the approaches of the "projection noise" experiment. Since "Andreff" utilizes a linear formulation it is quite fast. It is difficult to compare the speed of "Heller11", since it is implemented in MATLAB. However, as "Heller11" begins with the rotation estimation proposed by Park and Martin [6], which almost consumes the same time as "Andreff", we are convinced that "Heller11" should be slower than "Andreff" and "Ours". All in all, "Ours" runs slightly faster than "Andreff" and the slowest algorithms are "Heller11" and "Heller12". This result meets our expectations, since "Ours" is similar to a global SFM pipeline [36], which is regarded to be less robust but faster than the incremental SFM pipeline used in "Andreff", while "Ours" dispenses with global camera pose estimation. "Heller12" is a branch-and-bound search method, which doesn't need an initial guess, but its computation will grow exponentially with higher expected precision.

[1] To balance the accuracy and runtime of "Heller12", we set σ_{min} to 0.0001 in this experiment.

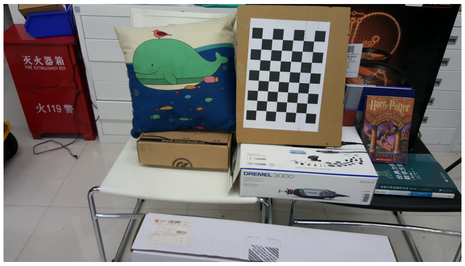

Fig. 4. Example photo from the monocular experiment.

TABLE III
ERROR AND RUNTIME COMPARISON FOR THE MONOCULAR EXPERIMENT.

Approach	Andreff	Heller11	Heller12[2]	Ours
Rotation Error	0.00499	0.00544	0.00642	**0.000409**
Translation Error(m)[3]	0.0119	0.0310	0.00860	**0.000648**
Runtime[4]	18.35s	nearly 6h	1038s	**15.55s**

B. Monocular Experiment

It is difficult to obtain the accurate transformation between a camera and another device, so there is no ground truth available to validate algorithms for extrinsic calibration relative to a camera. Indeed, this problem occurs in hand-eye calibration. In this experiment we assume that we are calibrating "two" cameras using the hand-eye calibration approach. But actually the two cameras are the same one: we set up a special scene including a chess pattern, like Fig. 4, and afterwards we move the camera and take a series of photos, ensuring that the entire chess pattern is in the view in every frame. Afterwards, we calibrate the camera using a chess pattern calibration approach to determine the camera poses, and regard this camera as the "hand". For the "eye" poses we assume that we don't know that there is a chess pattern and utilize hand-eye calibration approach to determine the transformation between "hand" and "eye". Obviously, the ground truth is the identity matrix for the rotation and a zero vector as translation. In this way, we overcome the problem that no ground truth is available.

In practice, we use the rear camera of a Samsung Galaxy S6 smart phone, with its AutoFocus disabled to satisfy our assumption that the camera is intrinsically calibrated, and a resolution of 3264×1836 pixels. Each square of the chess pattern is $28 \times 28mm$. We took 18 images in total at various directions and distances.

Table III illustrates the estimation errors and runtime of the four approaches. Regarding the estimation error, the accuracy of our approach is more than one order of magnitude better than the other algorithms, both in rotation and translation. Also, our approach is the fastest of all methods.

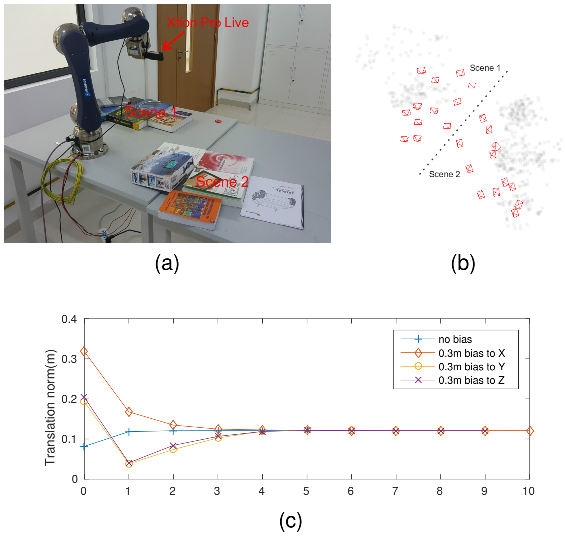

Fig. 5. Gripper-camera experiment. (a) set-up of gripper-camera calibration experiment. (b) reconstruction result. (c) the norm of final translation after biasing the translation estimated by initial hand-eye calibration.

C. Gripper-Camera Experiment

A Schunk LWA 4P lightweight robot arm with a two-finger gripper is utilized in this experiment, and we let the gripper grip an Asus Xtion Pro Live rigidly, as shown in Figure 5(a). Even though the Xtion Pro Live has a depth sensor, only the RGB camera is used here and the resolution is set to 1280×1024. We use the ROS MoveIt! software package along with the Schunk canopen driver to control the arm and get the hand poses with respect to the robot base.

Figure 5(a) also shows two independent scenes deliberately made by us, aiming to test the feasibility to merge the reconstructions for them using our approach. 24 motions are set manually, half of whom face scene 1 and the others face scene 2. 24 pairs of gripper poses and the corresponding images are obtained during this procedure. No images depicting both scene 1 and scene 2 were taken and thus SFM algorithms cannot create a single model containing both scenes.

Finally, our approach is performed, and the reconstruction result is given in Figure 5(b). It should be noted that the reconstructions of scene 1 and scene 2 are merged because the hand-eye calibration is integrated into the bundle adjustment.

Since no true gripper-camera transformation is available, we measure the translation from the gripper to the camera by hand with the known mechanical structure of the gripper and Xtion Pro Live, which is approximated to $[-0.020, 0.000, -0.107]^T$. This corresponds to the approximate 10cm distance between the camera and the center of the other side of the gripper and the fact that the RGB camera was approximately 2cm off center. The estimated translation by our approach is $[-0.0211, -0.00117, -0.119]^T$, very close to the manual measurement.

To evaluate the robustness of the bundle adjustment, we corrupt the translation result, which is estimated by the initial hand-eye calibration, by adding 0.3m to the X, Y or Z component, respectively. Figure 5(c) illustrates the norm

[2] $\sigma_{min} = 0.001$ in this experiment.

[3] Since the ground truth is zero, the translation error is determined by the norm of estimated translation rather than Eq. (25).

[4] The runtime of the monocular experiment is much longer than synthetic-data experiment, this results from extra feature extraction and matching which are very time-consuming.

(length) of the estimated translation after each iteration. One can see that, although the initial guess is corrupted heavily, the calibration results after bundle adjustment still converge, and the norm difference to no bias is less than $10^{-5}m$. This experiment indicates that our algorithm is very robust. Of course, more iterations are needed if the initial guess is corrupted.

VII. Conclusion

In this paper, we present a novel pipeline for hand-eye calibration. We first propose a hand-eye calibration algorithm based on 2-view matches as the initial guess. Afterwards bundle adjustment for simultaneous hand-eye calibration and reconstruction is performed to obtain a more accurate result. It should be noted that our algorithm is an extended hand-eye calibration solver, *i.e.* the calibration can be performed solely from a natural scene without any known object. Moreover, our approach is not limited to a set of connected views, and can even merge two or more reconstructions. Most importantly, synthetic and real data experiments indicate that our method is superior to existing algorithms in both accuracy and speed of the hand-eye calibration.

References

[1] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, June 1989.

[2] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, "An invitation to 3-d vision: from images to geometric models," 2004.

[3] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[4] J. C. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *The International Journal of Robotics Research*, vol. 10, no. 3, pp. 240–254, 1991.

[5] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, Feb. 1989.

[6] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving AX=XB on the Euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, Oct. 1994.

[7] K. Daniilidis, "Hand-eye calibration using dual quaternions," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.

[8] N. Andreff, R. Horaud, and B. Espiau, "On-line hand-eye calibration," in *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*. IEEE, 1999, pp. 430–436.

[9] H. Zhuang and Y. C. Shiu, "A noise-tolerant algorithm for robotic hand-eye calibration with or without sensor orientation measurement," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 4, pp. 1168–1175, 1993.

[10] R. Horaud and F. Dornaika, "Hand-eye calibration," *The International Journal of Robotics Research*, vol. 14, no. 3, pp. 195–210, 1995.

[11] Z. Zhao, "Hand-eye calibration using convex optimization," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 2947–2952.

[12] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617–622, 1998.

[13] J. Schmidt, F. Vogt, and H. Niemann, "Calibration–Free Hand–Eye Calibration: A Structure–from–Motion Approach," in *Pattern Recognition*. Springer Berlin Heidelberg, Aug. 2005, no. 3663, pp. 67–74.

[14] J. Heller, M. Havlena, A. Sugimoto, and T. Pajdla, "Structure-from-motion based hand-eye calibration using L∞ minimization," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3497–3503.

[15] J. Heller, M. Havlena, and T. Pajdla, "A branch-and-bound algorithm for globally optimal hand-eye calibration," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 1608–1615.

[16] T. Ruland, T. Pajdla, and L. Krger, "Globally optimal hand-eye calibration," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 1035–1042.

[17] J. Heller, M. Havlena, and T. Pajdla, "Globally Optimal Hand-Eye Calibration Using Branch-and-Bound," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 1027–1033, May 2016.

[18] H. Zhuang, Z. S. Roth, and R. Sudhakar, "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form AX= YB," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 549–554, 1994.

[19] R. L. Hirsh, G. N. DeSouza, and A. C. Kak, "An iterative approach to the hand-eye and base-world calibration problem," in *IEEE International Conference on Robotics and Automation, 2001. Proceedings 2001 ICRA*, vol. 3, 2001, pp. 2171–2176 vol.3.

[20] M. Shah, "Solving the robot-world/hand-eye calibration problem using the Kronecker product," *Journal of Mechanisms and Robotics*, vol. 5, no. 3, p. 031007, 2013.

[21] A. Li, L. Wang, and D. Wu, "Simultaneous robot-world and hand-eye calibration using dual-quaternions and Kronecker product," *International Journal of Physical Sciences*, vol. 5, no. 10, pp. 1530–1536, 2010.

[22] A. Malti, "Handeye calibration with epipolar constraints: application to endoscopy," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 161–169, 2013.

[23] L. Wu, J. Wang, L. Qi, K. Wu, H. Ren, and M. Q.-H. Meng, "Simultaneous Hand-Eye, Tool-Flange, and Robot-Robot Calibration for Comanipulation by Solving the AXB=YCZ Problem," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 413–428, 2016.

[24] M. K. Ackerman and G. S. Chirikjian, "A probabilistic solution to the AX=XB problem: Sensor calibration without correspondence," in *Geometric Science of Information*. Springer, 2013, pp. 693–701.

[25] M. K. Ackerman, A. Cheng, and G. Chirikjian, "An information-theoretic approach to the correspondence-free AX=XB sensor calibration problem," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 4893–4899.

[26] Q. Ma, H. Li, and G. S. Chirikjian, "New probabilistic approaches to the AX = XB hand-eye calibration without correspondence," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4365–4371.

[27] H. Li, Q. Ma, T. Wang, and G. S. Chirikjian, "Simultaneous Hand-Eye and Robot-World Calibration by Solving the Problem Without Correspondence," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 145–152, Jan. 2016.

[28] Q. Ma, Z. Goh, and G. S. Chirikjian, "Probabilistic approaches to the AXB=YCZ calibration problem in multi-robot systems," in *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[29] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, *Bundle Adjustment — A Modern Synthesis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 298–372.

[30] H. Neudecker, "Some theorems on matrix differentiation with special reference to kronecker matrix products," *Journal of the American Statistical Association*, vol. 64, no. 327, pp. 953–963, 1969.

[31] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[32] C. Sweeney, "Theia multiview geometry library: Tutorial & reference," http://theia-sfm.org.

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[34] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[35] J.-Y. Bouguet, "Camera Calibration Toolbox for Matlab," http://www.vision.caltech.edu/bouguetj/calib_doc/index.html, 2004.

[36] P. Moulon, P. Monasse, and R. Marlet, "Global fusion of relative motions for robust, accurate and scalable structure from motion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3248–3255.