Image and Vision Computing 70 (2018) 46–54

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Marker-based non-overlapping camera calibration methods with additional support camera views☆

Fangda Zhao*, Toru Tamaki*, Takio Kurita, Bisser Raytchev, Kazufumi Kaneda

Graduate School of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima City, Hiroshima 739-8527, Japan

ARTICLE INFO

Article history:
Received 24 February 2017
Received in revised form 19 October 2017
Accepted 25 December 2017

Keywords:
Non-overlapping camera calibration
AR marker
Pose estimation

ABSTRACT

Simple methods to calibrate non-overlapping cameras using markers on the cameras are proposed. By adding an augmented reality (AR) marker to a camera, we can find the transformation between the fixed AR marker and the camera. With such information, the relative pose between cameras can be found as long as the markers are visible to additional support cameras. The proposed method consists of two steps: (1) use of an extra support camera and a chessboard to find the transformation between the AR marker and the camera and (2) use of the transformation between markers to calibrate non-overlapping cameras. Compared to an existing method, the proposed method works stably and uses fewer images.

1. Introduction

Camera calibration has been investigated for a long time and remains a popular topic [1,2]. Most vision algorithms, for example [3–5], require accurate intrinsic camera parameters and, if multiple cameras are used, extrinsic camera parameters [4].

Stereo camera calibration [6] is the most common case that requires extrinsic parameter estimation. Stereo cameras necessarily share the field of view (FOV); therefore, objects and markers in the shared FOV can be used to estimate the extrinsic camera parameters. Many 3D reconstruction algorithms have benefited from this type of calibration process [5].

Another type of extrinsic calibration, i.e., non-overlapping camera calibration [7], does not share the FOV, thus making stereo calibration methods inapplicable. Due to different cost and accuracy requirements, different non-overlapping camera-calibration methods have been proposed for different applications, such as surveillance [8] and autonomous vehicle navigation [9]. However, none of these methods is universal, and each has advantages and disadvantages, as we describe in Section 1.1.

In this paper, we propose portable and stable calibration methods for non-overlapping cameras using markers.[1] The basic concept is rather straightforward, i.e., we estimate the transformation between multiple cameras by estimating the transformation between markers. Specifically, we place a marker on each camera to be calibrated (i.e., *target* cameras). Then, we capture images of the target cameras using other cameras (i.e., *support* cameras) such that all target camera markers are captured simultaneously in the support cameras' FOVs. An overview of the proposed method's configuration is shown in Fig. 1. Here, the task is to estimate the transformation between the target cameras (denoted by 1 and 2 in Fig. 1) using the support cameras. The proposed method consists of two parts. First, we use the support cameras and calibration (chessboard) patterns to estimate transformations between each target camera and corresponding marker pairs. Then, we estimate the transformations between the target cameras.

Our primary contributions are as follows.

- We present methods to calibrate the extrinsic parameters of non-overlapping cameras using external support cameras with markers.
- The proposed methods are evaluated using synthetic and real data to demonstrate their robustness against various camera configurations.

☆ This paper has been recommended for acceptance by Hongdong Li.

* Corresponding authors.
E-mail addresses: zhao@eml.hiroshima-u.ac.jp (F. Zhao), tamaki@hiroshima-u.ac.jp (T. Tamaki).

[1] A conference version of this paper was presented [10]. This paper extends that version by adding two different methods to the preparation step and providing additional experimental results.

https://doi.org/10.1016/j.imavis.2017.12.006
0262-8856/

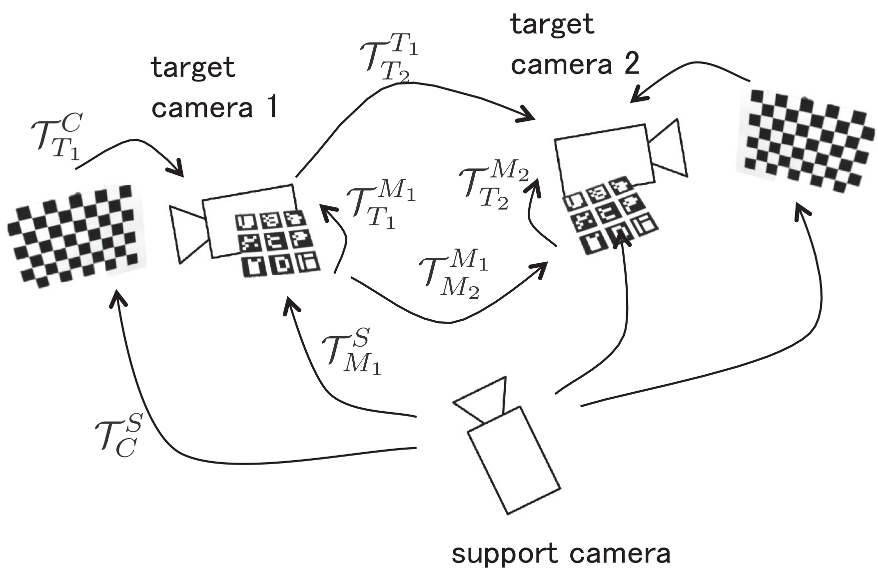


Fig. 1. Configuration overview.

The remainder of this paper is organized as follows. In the following subsection, we review related work. We describe the proposed method in Sections 2 and 3. We then provide and evaluate experimental results in Section 4. Finally, conclusions and suggestions for future work are presented in Section 5.

1.1. Related work

We categorize previous work on non-overlapping calibration into trajectory-, SLAM-, mirror-, and tracking-based methods. Trajectory- and SLAM-based methods focus on calibrating cameras attached to a mobile camera rig, whereas mirror- and tracking-based methods focus on calibrating a set of stationary cameras.

In trajectory-based methods [9,11], the transformation between fixed rig cameras should not change regardless of how the camera rig moves. Each camera on a moving rig captures image sequences, and camera motions (trajectories) are estimated for each camera. Then, the method attempts to compute extrinsic camera parameters by matching the camera trajectories. However, matching trajectories can suffer from degenerated cases, e.g., when motion occurs along a straight line. This restricts trajectory-based methods to applications with a large working space where complex camera motions can occur. In addition, accuracy is affected by the camera-motion estimation, which can be unstable in some cases. Furthermore, methods of this kind [11,12] need non-overlapping cameras to be fixed on a mobile rig. Such a requirement would be reasonable if cameras were on a vehicle for autonomous driving, but in some applications cameras are expected to be fixed on static solid walls, pillars or ground surface so that cameras mounted on them cannot be moved. Such a situation limits the usage of methods of this type.

SLAM-based methods [13,14] benefit from recent advancements in SLAM methods [3,15]. If we have a large calibration pattern that covers all non-overlapping cameras' FOVs, we can theoretically obtain the extrinsic parameters. However, generating such a large calibration pattern is impractical. By contrast, SLAM 3D reconstruction of a scene can be used as a type of calibration pattern, and transformation between cameras can be obtained by registering the 3D geometries of the same scene reconstructed from each camera. SLAM-based methods do not suffer from degenerate cases; however, scenes must be initialized in advance. Combinations of SLAM- and trajectory based methods that employ 3D information to handle degenerate cases associated with trajectory-based have been proposed [12].

The idea behind mirror-based methods [7,16,17] is interesting. The main difficulty with non-overlapping calibration is non-overlapped FOVs. However, a mirror can be used to reflect a single calibration pattern to all cameras, which makes it possible to estimate the transformation between the camera and the reflected calibration pattern. Since the camera and the calibration pattern are fixed, the pose of the mirror can also be estimated. Poses between cameras can be obtained by computing the transformations between the pattern and cameras individually. However, with mirror-based methods, it is difficult to handle mirror positions to obtain good and stable accuracy. It is known that calibration patterns should be captured in an image screen sufficiently large to increase accuracy; however, this is difficult because the results are sensitive to mirror poses, and there are various degenerated mirror poses. Therefore, mirror poses must be arranged carefully. In addition, mirror poses must satisfy a physical constraint of the mirror–camera configuration in the working space of the real environment.

Tracking-based methods [8,18] attempt to track the same object in 3D scenes captured by different cameras. These methods are typically applied to camera networks, such as surveillance systems, with multiple cameras. The motion between cameras can be predicted using a Kalman filter, and different camera poses can be estimated. Typically, tracking-based methods do not have high accuracy; however, they may be the easiest approach to calibrating relative poses of a large camera network.

We compare the existing methods in Table 1, which shows that for stationary camera calibration, existing methods suffer from either degenerate cases or do not have sufficiently high accuracy. Therefore, we propose a highly accurate method that does not suffer from degenerate cases. Experimental results indicate that the proposed method demonstrates state-of-the-art accuracy and high stability. Thus, it is expected that the proposed method is suitable for situations in which existing methods are inappropriate.

2. Formulation

The proposed method involves two steps: 1) finding the transformation between a target camera and an augmented reality (AR)

Table 1
Comparison of methods.

	Trajectory	SLAM	Mirror	Tracking	Proposed
Mobile cameras	Necessary	Necessary	Not necessary	Not necessary	Not necessary
Degenerate cases	Yes	No	Yes	No	No
Initialization	Not necessary	Necessary	Not necessary	Not necessary	Not necessary
Additional staff	No	No	Mirror	No	Marker and support camera
Accuracy	High	High	High	Low	High

marker [19] and 2) using this information to find the transformation between non-overlapping cameras. First, we define coordinate systems and transformations between cameras, calibration patterns, and markers.

Assume N_t denotes target cameras $T_i (i = 1, 2, \ldots, N_t)$ and N_s denotes support cameras $S_j (j = 1, 2, \ldots, N_s)$. Note that a single support camera is sufficient to explain the concept of the proposed method; however, we can obtain multiple (i.e., N_s) different viewpoints by moving a single support camera. In addition, we assume that an AR marker M_i is attached to each target camera and that calibration patterns (e.g., the plane of a chessboard pattern) $C_k (k = 1, 2, \ldots, N_c)$ are placed in the scene as a reference. The goal of the task is to estimate the transformations between target cameras or, equivalently, between the first target camera and the remaining target cameras. Hereafter, we use T_i, S_j, M_i, and C_k to denote corresponding coordinate systems, i.e., T_1 is the coordinate system of the first target camera, and so on.

We denote $T_{T_j}^{T_i} = \left(R_{T_j}^{T_i}, \mathbf{t}_{T_j}^{T_i}\right) \in SE(3)$ as a coordinate system transformation from T_i to T_j. In other words, if point $\mathbf{X} \in \mathbb{R}^3$ is represented as homogeneous coordinates $\mathbf{X}_{T_i} \in \mathcal{P}^3$ in T_i, the following equation holds:

$$\mathbf{X}_{T_j} = \mathcal{T}_{T_j}^{T_i} \mathbf{X}_{T_i}, \tag{1}$$

Therefore, we have $\mathcal{T}_{T_i}^{T_j} = \left(\mathcal{T}_{T_j}^{T_i}\right)^{-1}$. With this notation, the goal is to estimate $\left\{\mathcal{T}_{T_1}^{T_j}\right\}_{j=2}^{N_t}$ based on observations of AR markers and chessboard patterns.

To simplify the discussion, we assume that there are $N_t = 2$ target cameras, an $N_s = 1$ support camera, and an $N_c = 1$ calibration pattern. However, the discussion is applicable to other generalized cases.

2.1. Step 1: transformation between a target camera and a marker

First, we must obtain transformation $\mathcal{T}_{T_i}^{M_i}$ between target camera T_i and marker M_i in the scene captured by the support camera S_1. The AR marker detector [20] allows us to obtain the transformations $\mathcal{T}_{S_1}^{M_i}$. In addition, by placing a calibration pattern in the views of both the target and support cameras, as shown in Fig. 2 (a), we can employ a standard extrinsic camera calibration method [1] to obtain transformations $\mathcal{T}_{T_i}^{C_1}$ and $\mathcal{T}_{S_1}^{C_1}$ from the calibration pattern C_1 to the cameras T_i and S_1.

At this point, we have the following:

$$\mathcal{T}_{T_i}^{M_i} = \mathcal{T}_{T_i}^{C_1} \mathcal{T}_{C_1}^{S_1} \mathcal{T}_{S_1}^{M_i}. \tag{2}$$

Note that T_1 and T_2 can use different S_1 and C_1. In other words, different sets of support cameras and calibration patterns can be used for different target cameras.

The assumption that the calibration pattern is visible from both the target and support cameras seems to be strong; however, it is still practical. The calibration pattern is only required to obtain $\mathcal{T}_{T_i}^{M_i}$, and, if the assumption is satisfied, we can freely move the target camera, support cameras, and the calibration pattern.

Note that we perform the above procedure for each target camera. Once the target cameras are calibrated, we can place the target cameras on a rig in a configuration in which they do not share views and the markers remain attached to the target cameras.

2.2. Step 2: transformation between target cameras

Next, we capture the scene of the target cameras that do not overlap the support camera scene such that markers appear in the support camera view, as shown in Fig. 2 (b). Using the AR marker detector, we obtain $\mathcal{T}_{S_1}^{M_i}$. Note that here, the S_1 value differs from that in the previous subsection. By combining $\mathcal{T}_{T_i}^{M_i}$, we obtain the transformation between the target cameras as follows:

$$\mathcal{T}_{T_1}^{T_2} = \mathcal{T}_{T_1}^{M_1} \mathcal{T}_{M_1}^{S_1} \mathcal{T}_{S_1}^{M_2} \mathcal{T}_{M_2}^{T_2}. \tag{3}$$

3. Proposed method

The process described above does not consider the error in each component transformation; therefore, it is inherently not robust to noise. To reduce the effect of noise, a natural solution is to use multiple observations, i.e., multiple support cameras S_j and calibration patterns C_k.

Here, we assume that transformations $\left\{\mathcal{T}_{T_i}^{C_k}\right\}$, $\left\{\mathcal{T}_{M_i}^{S_j}\right\}$, and $\left\{\mathcal{T}_{S_j}^{C_k}\right\}$ obtained in the preparation step are reasonably accurate. Thus, they are considered as given and fixed in the calibration step (Section 3.2) to estimate $\mathcal{T}_{T_1}^{T_2}$. The methods used in the preparation step (Section 3.1) have been well studied and can be considered optimal in a local configuration, i.e., between the cameras and the calibration pattern or markers.

Based on this assumption, we introduce the proposed method for the preparation and calibration steps.

3.1. Preparation step

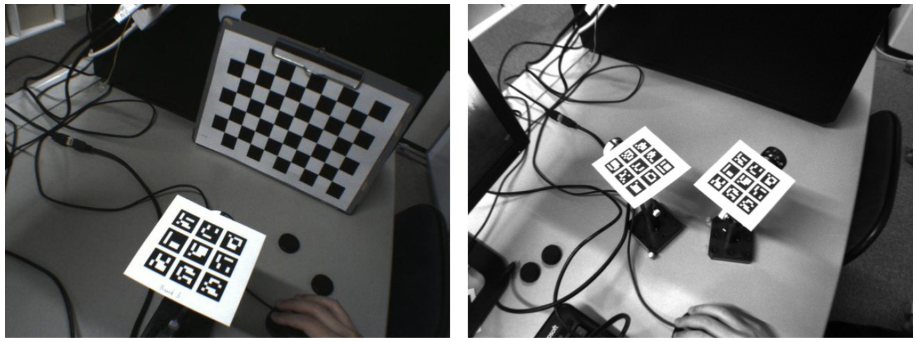
To estimate transformation[2] $\mathcal{T}_T^M$, multiple observations may be acquired by moving the support camera $\{S_j\}$ and the calibration pattern $\{C_k\}$.

We propose three methods of achieving this goal.

3.1.1. 3D point registration-based method

The first method compares the n-th point $X_{C_k,n}$ in the target camera coordinate system T to calibration patterns C_k transformed

[2] To simplify the notation, we omit index i.



(a) Preparation step (b) Calibration step

Fig. 2. Support camera view for preparation and calibration steps.

through two different paths, i.e., from C_k to T and from C_k to T through S_j and M. This leads to the following minimization problem:

$$\min_{\mathcal{T}_T^M} \sum_{k=1}^{N_C} \sum_{j=1}^{N_s} \sum_{n} E,$$
$$E = \left\| \mathcal{T}_T^{C_k} \mathbf{X}_{C_k,n} - \mathcal{T}_T^M \mathcal{T}_M^{S_j} \mathcal{T}_{S_j}^{C_k} \mathbf{X}_{C_k,n} \right\|^2. \tag{4}$$

In practice, this is equivalent to the following:

$$E = \left\| \mathbf{Y}_T^{j,k,n} - \left(R_T^M \mathbf{X}_M^{j,k,n} + \mathbf{t}_T^M \right) \right\|^2, \tag{5}$$

where

$$\mathbf{Y}_T^{j,k,n} = R_T^{C_k} \mathbf{X}_{C_k,n} + \mathbf{t}_T^{C_k} \tag{6}$$

$$\mathbf{X}_M^{j,k,n} = R_M^{S_j} \left(R_{S_j}^{C_k} \mathbf{X}_{C_k,n} + \mathbf{t}_{S_j}^{C_k} \right) + \mathbf{t}_M^{S_j}. \tag{7}$$

In other words, the problem is to estimate the transformation between two sets of corresponding 3D points $\left\{ \mathbf{Y}_T^{j,k,n}, \mathbf{X}_M^{j,k,n} \right\}$. This is a standard orthogonal Procrustes problem for which many solvers exist [21–23]. Note that $\mathbf{Y}_T^{j,k,n}$ is the same for all j.

3.1.2. Reprojection error-based method

Many calibration methods minimize reprojection errors [1,17,24], which is the second idea, i.e., minimizing the error between the detected 2D points of calibration pattern $c_{k,n}$ in the target camera's image and the re-projected calibration pattern $c'_{k,n,j}$.

We attempt to minimize the following cost function:

$$\min_{\mathcal{T}_T^M} \sum_{k=1}^{N_C} \sum_{j=1}^{N_s} \sum_{n} E,$$
$$E = \left\| c_{k,n} - c'_{k,n,j} \right\|^2. \tag{8}$$

The re-projected point of the calibration pattern $c'_{k,n,j}$ is obtained as follows:

$$c'_{k,n,j} = f_T \left(\mathcal{T}_T^M \mathcal{T}_M^{S_j} \mathcal{T}_{S_j}^{C_k} \mathbf{X}_{C_k,n} \right) = f_T \left(\mathcal{T}_T^M \mathbf{X}'_{C_k,n} \right) \tag{9}$$

where $f_T()$ is the projection of camera T and $\mathbf{X}'_{C_k,n} = \mathcal{T}_M^{S_j} \mathcal{T}_{S_j}^{C_k} \mathbf{X}_{C_k,n}$.

This is a pose-estimation problem that can be solved by the Levenberg–Marquardt algorithm. Note that we use the Ceres solver [25] implementation of this algorithm.

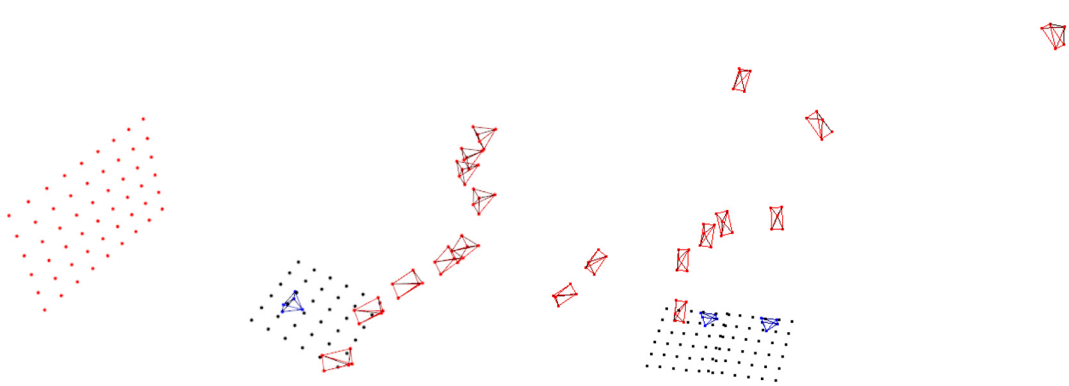


Fig. 3. Configurations of the synthetic datasets for (left) the preparation step and (right) the calibration step. Support and target cameras are shown in red and blue, respectively. The red grid is a calibration pattern, and the black grids are markers on the target cameras.

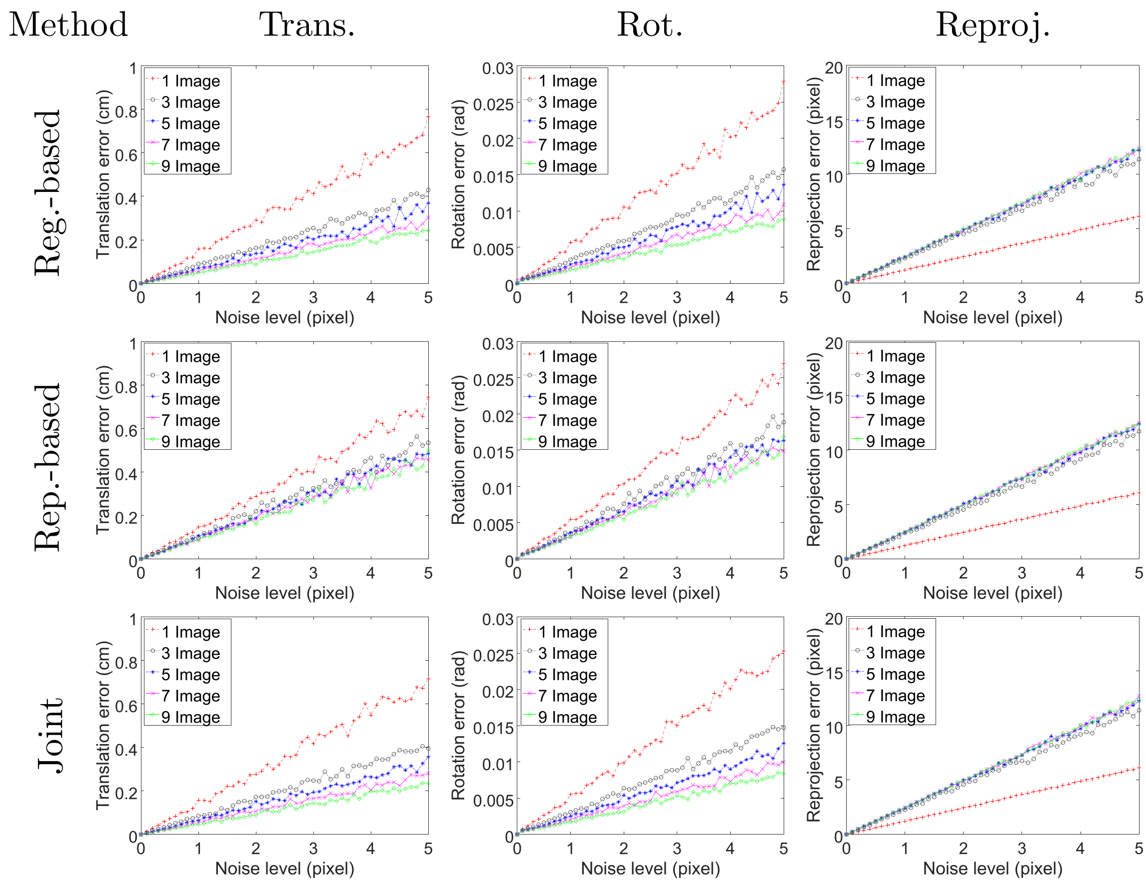


Fig. 4. Performance comparison of registration-based (first row), reprojection-error-based (second row), and joint-optimization (third row) methods: error in estimating translation vectors (first column), rotation angles of $\mathcal{T}_T^M$ (second column), and reprojection errors on the target camera's image (third column).

3.1.3. *Reprojection error-based joint optimization*

The third idea is to find all required transformations in a large joint optimization problem to minimize reprojection errors. In other words, we attempt to minimize the following cost function:

$$\min_{\mathcal{T}_T^M, \mathcal{T}_{S_j}^{C_k}, \mathcal{T}_{S_j}^M, \mathcal{T}_T^{C_k}} E,$$

$$E = \sum_{k,j,n} \left\| f'\left(\mathcal{T}_{S_j}^{C_k} \mathbf{X}_{C_k,n}\right) - c_{k,n,j} \right\|^2 + \sum_{j,n'} \left\| f'\left(\mathcal{T}_{S_j}^{M} \mathbf{X}_{M'_n}\right) - m_{j,n'} \right\|^2 + \sum_{k,n} \left\| f\left(\mathcal{T}_{T}^{C_k} \mathbf{X}_{C_k,n}\right) - c_{k,n} \right\|^2 + \sum_{k,j,n'} \left\| f\left(\mathcal{T}_{T}^{M} \mathcal{T}_{M}^{S_j} \mathcal{T}_{S_j}^{C_k} \mathbf{X}_{C_k,n}\right) - c'_{k,n,j} \right\|^2, \quad (10)$$

where $m_{j,n'}$ is the n'-th 2D marker point in the support camera's view and $X_{M',n}$ is the n-th 3D point on marker M'. From the first to the last terms, we attempt to minimize the reprojection errors of points on the calibration patterns (first term), markers in the support cameras (second term), the calibration patterns in the target camera (third term), and the calibration patterns in the target camera through the support camera and the marker (fourth term).

Note that this optimization is complex and non-linear, and requires a good initial value. We use the results obtained from a setting with one support camera and calibration pattern ($N_s = N_c = 1$) as the initial value for this problem. In our experiments, we also use the Levenberg–Marquardt algorithm in the Ceres solver [25].

3.2. *Calibration step*

In the preparation step described above, we obtain transformations $T_{T_i}^{M_i}$ for all marker-target-camera pairs. In the calibration step, we obtain the transformation between target cameras.

Here, the transformation to be estimated is $T_{M_2}^{M_1}$. To this end, we propose an idea that is similar to that discussed in Section 3.1.1, i.e.,

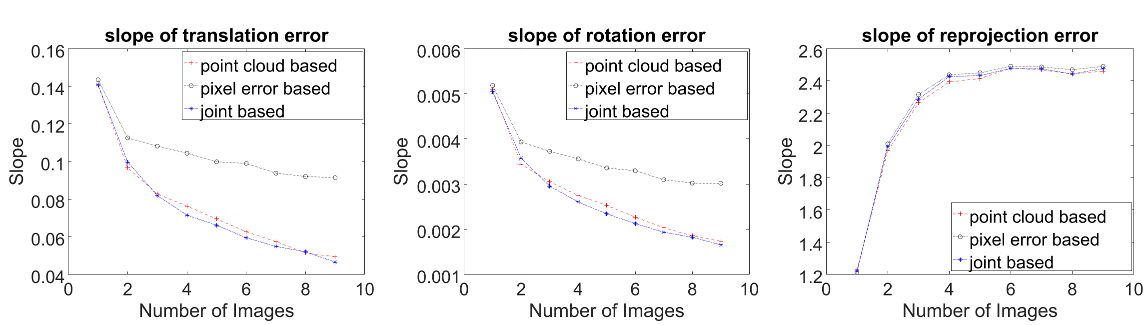


Fig. 5. Slopes of three methods relative to different error metrics.

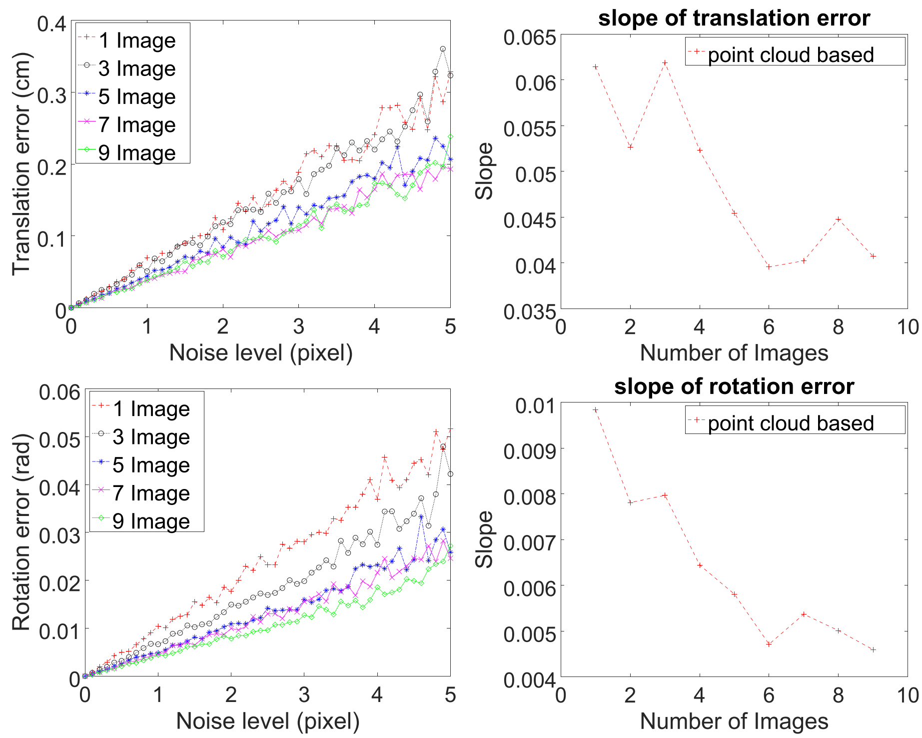


Fig. 6. Error in estimating (first row) translation vectors and (second row) rotation angles of $\mathcal{T}_{T_1}^{T_2}$ by the point-cloud registration-based method and slope of them.

we take advantage of multiple observations S_j by moving the support camera. The difference is the points used. We compare points $\mathbf{X}_{M_1,n}$ (n-th point of M_1 in the coordinate system M_2 transformed) through different transformation paths; M_1 to M_2 directly, and M_1 to M_2 through S_j. This is formulated as the following problem:

$$\min_{\mathcal{T}_{M_2}^{M_1}} \sum_{j=1}^{N_S} \sum_n E,$$

$$E = \left\| \mathcal{T}_{M_2}^{S_j} \mathcal{T}_{S_j}^{M_1} \mathbf{X}_{M_1,n} - \mathcal{T}_{M_2}^{M_1} \mathbf{X}_{M_1,n} \right\|^2. \tag{11}$$

Again, this can be written as a Procrustes problem as follows:

$$E = \left\| \mathbf{Y}_{M_2}^{j,n} - \left(R_{M_2}^{M_1} \mathbf{X}_{M_1}^{j,n} + \mathbf{t}_{M_2}^{M_1} \right) \right\|^2, \tag{12}$$

where

$$\mathbf{Y}_{M_2}^{j,n} = R_{M_2}^{S_j} \left(R_{S_j}^{M_1} \mathbf{X}_{M_1,n} + \mathbf{t}_{S_j}^{M_1} \right) + \mathbf{t}_{M_2}^{S_j} \tag{13}$$

$$\mathbf{X}_{T_2}^{j,n} = \mathbf{X}_{M_1,n}. \tag{14}$$

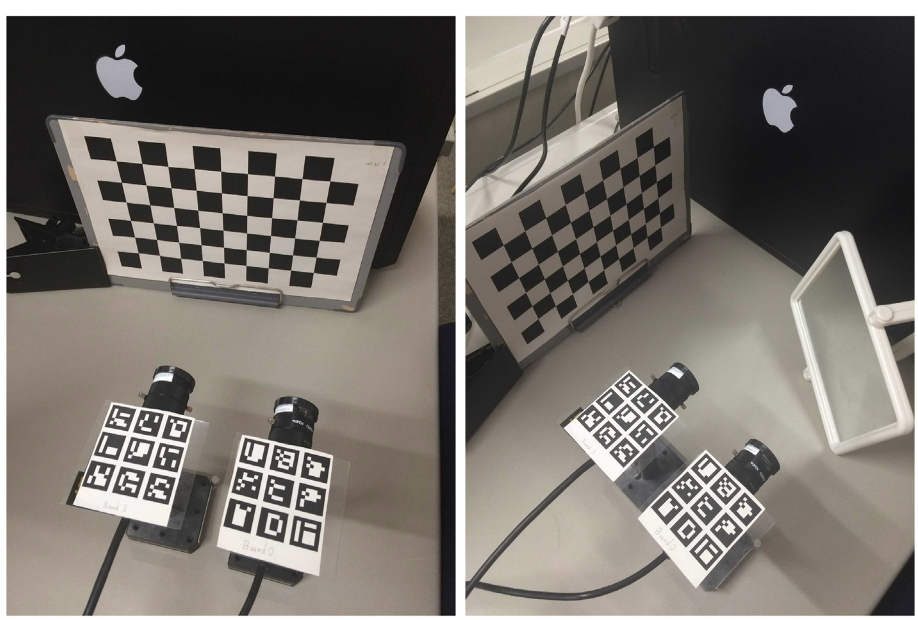


Fig. 7. Settings used to perform stereo calibration (left) and Takahashi et al.'s [17] method (right).

Finally, we obtain the transformation between target cameras as follows:

$$\mathcal{T}_{T_1}^{T_2} = \mathcal{T}_{T_1}^{M_1} \mathcal{T}_{M_1}^{M_2} \mathcal{T}_{M_2}^{T_2}. \tag{15}$$

4. Experimental results

We evaluated the proposed methods in two settings. First, we used synthetic data to evaluate the robustness of the proposed methods in noise over different numbers of observations. Second, we used real images from overlapping cameras, which allowed us to use a stereo calibration result as a "ground truth", to compare performance among existing mirror-based methods.

4.1. Synthetic data for the preparation step

We created synthetic data by projecting the points of the calibration patterns and markers onto the target and support camera image screens. The internal parameters of both the target and support cameras were fixed in advance as follows:

$$\mathcal{K} = \begin{bmatrix} 1100 & 0 & 800 \\ 0 & 1100 & 600 \\ 0 & 0 & 1 \end{bmatrix}. \tag{16}$$

The coordinate systems T, M, S_j, and C were arbitrarily placed in a 3D scene such that the calibration patterns were visible to both the target and support cameras (Fig. 3 visualizes this configuration). Here, we have $N_s = 9$ support camera images. We fixed the transformations of the coordinate systems and used them as the ground truth. Note that rotations are represented by quaternions to avoid the Gimbal lock effect.

We added Gaussian noise (zero mean and standard deviation σ as the noise level) onto the 2D coordinates of the projected points. We then obtained transformations by applying a standard perspective-n-point problem [26]. Then, we compared the estimated transformations $\mathcal{T}_T^M$ and $\mathcal{T}_{T_1}^{T_2}$ to the corresponding ground truth transformations.

Fig. 4 shows relative errors of $\mathcal{T}_T^M$, showing errors for estimated translation vector rotations, and average reprojection errors in the target camera, respectively, by averaging results of 100 trials for each noise level σ. Each row of Fig. 4 corresponds to three proposed methods, and each column shows errors in translation, rotation, as well as reprojection. In each graph, different plots show average errors with different numbers of support camera images used for estimation, in which average errors increase as noise level becomes large.

Errors in translation and rotation with one support camera image increase largely compared to errors with multiple support images. However, it is difficult to see the effect of increasing support images with these plots. Fig. 5 visually shows the effect. First, we fit a line ($y = ax$ without bias) to each of the error-noise level plots and obtain the slope a. In the plots of Fig. 5, the slope is shown over different numbers of support camera images. The first two graphs in Fig. 5 clearly show the effect of using multiple images. We see that the registration-based and reprojection-error-based joint optimization methods perform equally and better than the reprojection-error-based method. The registration-based method appears to be the best because it is computationally much more efficient than the joint optimization.

The last column in Fig. 4 shows reprojection errors. These plots look different from the results above; the errors with a single image are better, and the slope appears to increase with the number of images. Besides, the reprojection-error-based method outperforms the registration-based method in this metric. This is reasonable because the reprojection-error-based method minimizes the reprojection error, whereas the registration-based method does not involve it. The other two methods have better results in rotation and translation error; therefore, this might suggest that the reprojection error is not a good indicator for comparison in this simulation setting.

4.2. Synthetic data for the calibration step

We created additional synthetic data with two target cameras placed almost in parallel, as shown in Fig. 3 (right). The results are shown in Fig. 6. Note that we used the noise-free transformation T_T^M; thus, accuracy was affected by only the calibration step. The results on the left show that errors in translation and rotation were slightly greater than those of the preparation step; however, the errors were

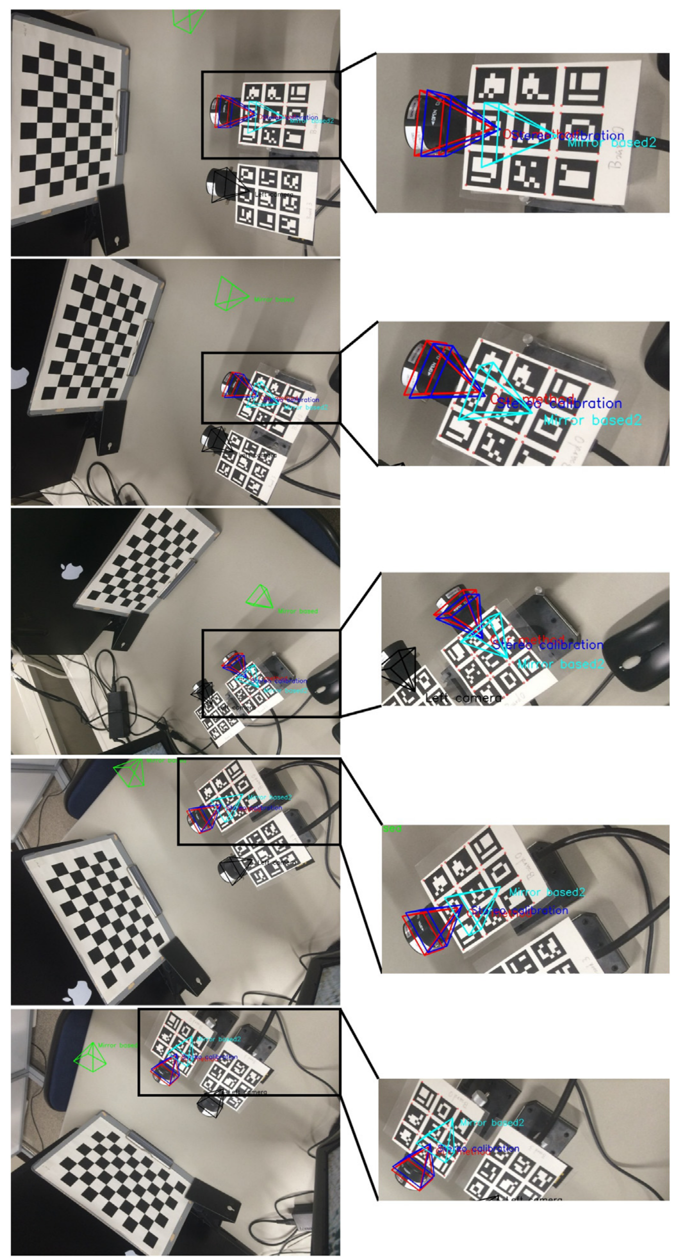


Fig. 8. Estimated camera poses in five images from the support camera under the stereo condition.

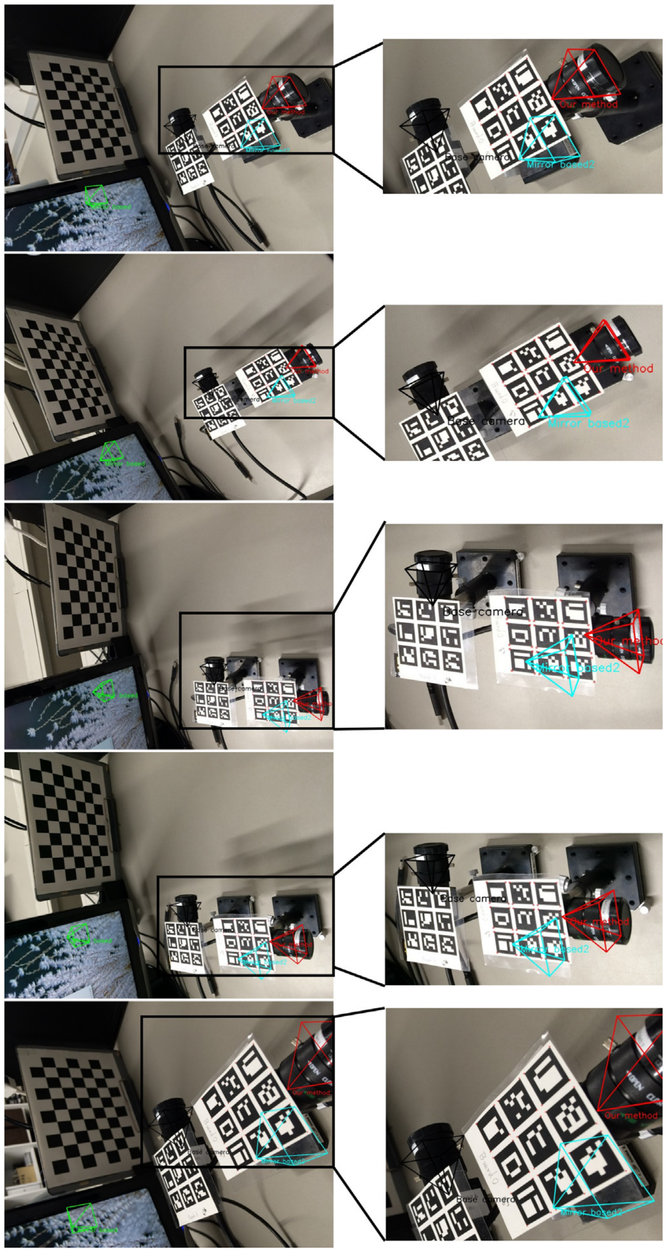


Fig. 9. Estimated camera poses in five images from the support camera under the non-overlapping condition.

still in acceptable ranges (up to 0.1 cm and 0.01 rad at a noise level of 2 pixels). The plots on the right show the slope of the error–noise-level plots, which indicate the effect of using multiple support camera images.

4.3. Real data

We used the proposed method to calibrate two USB3 cameras (PointGray FL3-U3-32S2C-CS, 1/2.8″ CMOS, max. resolution: 2080 × 1552) with SPACECOM L28CSWI lenses ($f = 2.8$ mm, F1.3 ∼ close (manual) and an angle of view of $92.0^\circ \times 71.7^\circ$, manual focus) using a support camera (iPhone 5s rear camera) with a default lens ($F = 2.2$, $f = 30$ mm). The image resolutions of the target cameras were 1600 × 1200 and 3264 × 2448 pixels for the support camera. To estimate $\mathcal{T}_{S_j}^{M_i}$, we used the ArUco board marker detector [20,27]. To estimate $\mathcal{T}_{T_i}^{C}$ and $\mathcal{T}_{S_j}^{C}$, we used an OpenCV [26] implementation of Zhang's method [1] with a chessboard calibration pattern. Note that the internal camera parameters were also estimated in this step.

Each of the cameras is fixed on a separate camera rig that is rigidly fixed on the table by a strong magnet. The baseline between two cameras is not so long, but this does not mean that our method is constrained by a small baseline. Our method works as long as the target markers are seen from the support camera.

For the performance evaluation, we compared the proposed method to standard stereo-camera calibration [6] implemented in OpenCV that directly estimates the transformation $\mathcal{T}_{T_1}^{T_2}$ between the target cameras. Note that this is possible only when two target cameras share their views; therefore, this setting is standard stereo calibration rather than non-overlapping calibration. However, we expect that our method should show similar performance in the non-overlapping situation.

We compare the proposed methods to mirror-based methods [17] and [28]. The mirror-based methods require that the calibration pattern be observed through mirrors. However, in our experimental setting, the calibration pattern was observed directly by both cameras. Therefore, we moved the calibration pattern such that the cameras could observe it through the mirror, while the relative camera positions were unchanged. Then, we obtained $T_{C_1}^{T_i}$ using code available online [29] to calculate transformation $T_{T_1}^{T_2}$. We also performed the same experiment in another setting with non-overlapping cameras to compare the results of the proposed and mirror-based methods without stereo calibration. Note that the mirror-based method [17] takes exactly three points and three images as input; therefore, we used all combinations of three points on the chessboard pattern to estimate the results and selected the result with the smallest reprojection error. The improved method [28] can accept more points and images; thus, we used five images and all points in a 9 × 6 chessboard pattern (Fig. 7).

For the proposed method, we took five images ($N_s = 5$) with the support camera and one image with each target camera to estimate $\mathcal{T}_{T_1}^{M_1}$ and $\mathcal{T}_{T_2}^{M_2}$. Note that the same images were used to estimate $\mathcal{T}_{T_1}^{T_2}$.

We visualized the estimated camera poses onto the support camera images for validation in Figs. 8 and 9. The pose of the left target camera T_1 estimated from T_{M1} is shown in black and indicated by "left camera" or "base camera". The right target camera T_2 obtained by the stereo calibration is shown in red. Note that this was considered to be the ground truth in this experiment.

The result of the proposed method (3D point registration-based) is shown in dark blue ("our method" in the images). As can be seen, the blue and red cameras are very close to each other, which means that the proposed method correctly estimated the transformation T_{T1}^{T2}. The results by [17] and [28] are shown in green ("mirror based") and cyan (" mirror based 2"), respectively. The results of our method appear to be similar to the stereo-calibration results and better than the results of the mirror-based methods. This may be due to the degenerated cases inherent to the mirror-based methods. Mirror-based methods require the configurations of cameras, mirrors, and calibration patterns to be limited in order to avoid unexpected failures encountered during actual setting shown here. By contrast, the proposed method has great configuration flexibility without hidden degenerated cases.

5. Conclusions

In this paper, we have proposed marker-based non-overlapping camera-calibration methods that can be used for both non-overlapping and overlapping cameras. By taking advantage of AR markers and support cameras, the proposed method first finds transformations between the AR markers and the target cameras. The proposed method works as long as the markers attached to the target cameras are visible to the support cameras.

We used both synthetic and real data to evaluate the proposed method. The results show that the 3D registration-based method demonstrates the best performance. Compared to other methods, our method is simpler and handles many images, thus improving accuracy, and does not suffer from degenerate cases evident in some existing methods. In future, we plan to apply the proposed method to real-world applications with non-overlapping cameras.

Acknowledgments

This work was supported in part by JSPS KAKENHI grant number JP26280061.

References

[1] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1330–1334. https://doi.org/10.1109/34.888718.

[2] S. Ramalingam, P. Sturm, A unifying theory for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. PP (99) (2016) 1-1. https://doi.org/10.1109/TPAMI.2016.2592904.

[3] C. Forster, M. Pizzoli, D. Scaramuzza, SVO: fast semi-direct monocular visual odometry, 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014. pp. 15–22. https://doi.org/10.1109/ICRA.2014.6906584.

[4] A. Geiger, J. Ziegler, C. Stiller, StereoScan: dense 3d reconstruction in real-time, 2011 IEEE Intelligent Vehicles Symposium (IV), 2011. pp. 963–968. https://doi.org/10.1109/IVS.2011.5940405.

[5] P.F. Alcantarilla, C. Beall, F. Dellaert, Large-scale dense 3D reconstruction from stereo imagery, 5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV13), 2013.

[6] R.I. Hartley, Theory and practice of projective rectification, Int. J. Comput. Vis. 35 (2) (1999) 115–127.

[7] R.K. Kumar, A. Ilie, J.-M. Frahm, M. Pollefeys, Simple calibration of non-overlapping cameras with a mirror, 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE. 2008, pp. 1–7. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4587676. https://doi.org/10.1109/CVPR.2008.4587676.

[8] B. Lamprecht, S. Rass, S. Fuchs, K. Kyamakya, Extrinsic camera calibration for an on-board two-camera system without overlapping field of view, IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. 2007, pp. 265–270. https://doi.org/10.1109/ITSC.2007.4357679.

[9] P. Lébraly, E. Royer, O. Ait-Aider, M. Dhome, Calibration of non-overlapping cameras - application to vision-based robotics, Proceedings of the British Machine Vision Conference, BMVA Press. 2010, pp. 10.1–10.12. https://doi.org/10.5244/C.24.10.

[10] F. Zhao, T. Tamaki, T. Kurita, B. Raytchev, K. Kaneda, Marker based simple non-overlapping camera calibration, 2016 IEEE International Conference on Image Processing (ICIP), 2016. pp. 1180–1184. https://doi.org/10.1109/ICIP.2016.7532544.

[11] S. Esquivel, F. Woelk, R. Koch, Calibration of a multi-camera rig from non-overlapping views, Joint Pattern Recognition Symposium, Springer. 2007, pp. 82–91.

[12] L. Heng, B. Li, M. Pollefeys, Camodocal: automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE. 2013, pp. 1793–1800.

[13] E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam, Y. Miki, Calibration of non-overlapping cameras using an external SLAM system, 2014 2nd International Conference on 3D Vision, vol. 1, IEEE. 2014, pp. 509–516. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7035864. https://doi.org/10.1109/3DV.2014.106.

[14] G. Carrera, A. Angeli, A.J. Davison, SLAM-based automatic extrinsic calibration of a multi-camera rig, 2011 IEEE International Conference on Robotics and Automation, IEEE. 2011, pp. 2652–2659. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980294. https://doi.org/10.1109/ICRA.2011.5980294.

[15] R.A. Newcombe, S.J. Lovegrove, A.J. Davison, DTAM: dense tracking and mapping in real-time, 2011 international conference on computer vision, IEEE. 2011, pp. 2320–2327.

[16] A. Agrawal, Extrinsic camera calibration without a direct view using spherical mirror, 2013 IEEE International Conference on Computer Vision, IEEE. 2013, pp. 2368–2375. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6751405. https://doi.org/10.1109/ICCV.2013.294.

[17] K. Takahashi, S. Nobuhara, T. Matsuyama, A new mirror-based extrinsic camera calibration using an orthogonality constraint, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012. pp. 1051–1058. https://doi.org/10.1109/CVPR.2012.6247783.

[18] N. Anjum, M. Taj, A. Cavallaro, Relative position estimation of non-overlapping cameras, 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, 2007. pp. II-281–II-284. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4217400. https://doi.org/10.1109/ICASSP.2007.366227.

[19] S. Garrido-Jurado, R.M. noz Salinas, F. Madrid-Cuevas, M. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recogn. 47 (6) (2014) 2280–2292. http://www.sciencedirect.com/science/article/pii/S0031320314000235. https://doi.org/10.1016/j.patcog.2014.01.005.

[20] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, M. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recogn. 47 (6) (2014) 2280–2292. http://www.sciencedirect.com/science/article/pii/S0031320314000235.

[21] K.S. Arun, T.S. Huang, S.D. Blostein, Least-squares fitting of two 3-D point sets, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-9 (5) (1987) 698–700. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4767965.

[22] K. Kanatani, Analysis of 3-D rotation fitting, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5) (1994) 543–549. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=291441.

[23] S. Umeyama, Least-squares estimation of transformation parameters between two point patterns, IEEE Trans. Pattern Anal. Mach. Intell. 13 (4) (1991) 376–380. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=88573.

[24] Y. Furukawa, J. Ponce, Accurate camera calibration from multi-view stereo and bundle adjustment, 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008. pp. 1–8. https://doi.org/10.1109/CVPR.2008.4587681.

[25] S. Agarwal, K. Mierle, Others, Ceres Solver, 2012, http://ceres-solver.org.

[26] G. Bradski, OpenCV libs, Dr. Dobb's Journal of Software Tools, 2000.

[27] S. Garrido-Jurado, R. Muñoz Salinas, F.J. Madrid-Cuevas, R. Medina-Carnicer, Generation of fiducial marker dictionaries using mixed integer linear programming, Pattern Recogn. 51 (2016) 481–491. http://www.sciencedirect.com/science/article/pii/S0031320315003544. https://doi.org/10.1016/j.patcog.2015.09.023.

[28] K. Takahashi, S. Nobuhara, T. Matsuyama, Mirror-based camera pose estimation using an orthogonality constraint, IPSJ Trans. Comput. Vis. Appl. 8 (2016) 11–19.

[29] K. Takahashi, S. Nobuhara, T. Matsuyama, OpenCV libs, A new mirror-based extrinsic camera calibration using an orthogonality constraint project website, 2012. https://computer-vision.github.io/takahashi2012cvpr/ accessed 2017/01/31.