Robust Plane-based Calibration of Multiple Non-Overlapping Cameras

Chen Zhu[1], Zihan Zhou[2], Zirang Xing[1], Yanbing Dong[1], Yi Ma[1], and Jingyi Yu[1,3]
[1]School of Information Science and Technology, ShanghaiTech University
[2]College of Information Sciences and Technology, The Pennsylvania State University
[3]Department of Computer and Information Sciences, University of Delaware
{zhuchen, xingzr, dongyb, mayi, yujy1}shanghaitech.edu.cn, zzhou@ist.psu.edu

Abstract

The availability of commodity multi-camera systems such as Google Jump, Jaunt, and Lytro Immerge have brought new demand for reliable and efficient extrinsic camera calibration. State-of-the-art solutions generally require that adjacent, if not all, cameras observe a common area or employ known scene structures. In this paper, we present a novel multi-camera calibration technique that eliminates such requirements. Our approach extends the single-pair hand-eye calibration used in robotics to multi-camera systems. Specifically, we make use of (possibly unknown) planar structures in the scene and combine plane-based structure from motion, camera pose estimation, and task-specific bundle adjustment for extrinsic calibration. Experiments on several multi-camera setups demonstrate that our scheme is highly accurate, robust, and efficient.

1. Introduction

With their tremendous applications in virtual reality, 3D mapping and robotic vision, multi-camera systems ranging from Point Grey's Ladybug cameras to Jaunt's 360 camera rig have become increasingly prevalent in fields of computer vision and robotics. Conducting precise and efficient calibration of extrinsic parameters across all cameras within such a system, that is, obtaining the relative poses of the cameras, is central for all such applications.

Approaches in Computer Vision. One possible solution is to "force" cameras to observe a common object, if not a common area. [13] proposed a technique where cameras only need to observe some part of the calibration board. The plane contains specially designed patterns amenable for feature extraction. Combined with planar geometric constraint, they can reliably extract the extrinsic parameters between pairs of cameras as long as they observe a part of the same plane but will fail when the cameras are positioned nearly back-to-back.

Other calibration tricks have also been proposed. [11] utilizes mirrors to create virtual views of the calibration board. Such a method is flexible for pairwise camera calibration but requires elaborate configuration and design. Further, when the number of cameras increases (e.g., a ring of cameras), it would require combinations of mirrors and the complexity of geometric model grows quickly. [5] packs the cameras close to each other and assumes same optical center to handle the most extreme case where no camera pairs have an overlapping FoV. Such a configuration eliminates translation from the extrinsics and therefore the resulting system cannot capture parallax.

It is also possible to exploit scene geometry for calibration, e.g., by matching 3D maps created by different cameras [4] and matching image features to a 3D map created by an external SLAM system [3]. As a simpler alternative, [10] makes use of the ground plane as common scene geometry and strategically position the camera so that they can uniformly see the ground. It is also worth noting that most of these previous approaches conduct pair-wise calibration, which undermines the accuracy of the estimated translations.

Approaches in Robotics. The robotics community has focused mainly on the hand-eye calibration problem where a camera ("eye") is mounted on the gripper ("hand") of a robot. The camera was calibrated using a calibration pattern. Then the unknown transformation from the robot coordinate system to the calibration pattern coordinate system as well as the transformation from the camera to the hand coordinate system need to be estimated simultaneously [1, 9, 8].

Most previous approaches exploit motion rigidity and 3D scene geometry. Since the "hand" cannot capture images, the problem hence resembles calibrating two non-overlapping cameras. [6] acquires camera trajectories through structure-from-motion (SfM), and as long as the

motion is non-planar, it can reliably extract the extrinsic parameters through optimization. [12] further extends [6] to handle most planar and non-planar motion by iteratively refining the trajectory, extrinsic parameters and the 3D scene points. Yet, all these approaches are applicable to two-camera systems whereas we focus on a more general configuration.

Contributions of this Paper. We present a novel hand-eye calibration technique for calibrating multi-camera systems with non-overlapping FoVs. Unlike previous methods, we do not assume the scene to have specific patterns (e.g., checkerboards) or known structures (e.g., external 3D maps). Instead, we propose to explore the (possibly unknown) planar surfaces in the scene for calibration. Our method can be applied to any textured scene planes. In particular, in this paper we utilize a calibration target composed of planar facades of random patterns [13] as well as generic planar surfaces in outdoor street views to demonstrate the generality of our method.

We start with conducting a plane-based SfM (e.g., [17]). Specifically, we extract feature points from the scene planes by SURF [2] and only match the feature points that have a unique portfolio in feature space. Next, we group the matched feature points across multiple frames into a trajectory. Note that, the planar constraint and trajectory formulation provide a strong evidence to exclude the mismatches between similar feature points, which often occurs when multiple similar calibration targets are used to calibrate multiple camera systems.

For motion estimation, we first detect and estimate the homography matrix of a plane through a consensus algorithm which preserves the best homography matrix estimated from the inlier trajectories. The relative camera poses are then estimated by decomposing the homography matrices. Finally, we conduct task-specific bundle adjustment to refine the extrinsic parameters, camera motions and coordinates of the feature points by minimizing the reprojection error in each frame. We experiment our technique on several multi-camera systems including a two-camera fisheye panorama system, a 16-camera circular array, and a five-camera system. Extensive experiments show that our solution outperforms the state-of-the-art techniques in accuracy, robustness, and efficiency.

In summary, our main contributions are as follows:

- To the best of our knowledge, we are the first to apply plane-based SfM to extrinsic calibration of multi-camera systems. Our method does not require any specific patterns or known scene structures, and obtains accurate results even in the cases when existing methods fail (i.e., due to image noises, degenerate scene configurations, or excessive manual interventions).
- To handle mismatches in the plane-based SfM, we propose a new method to detect outlying feature points based on their accumulative reprojection error across all frames. We further develop a novel task-specific bundle adjustment procedure to jointly refine the relative poses of all cameras and update the sets of outliers.
- As planes are prevalent in man-made scenes, our algorithm provides a viable solution to the calibration problem in a wide variety of real scenarios. In this paper, we conduct experiments on different multi-camera systems to demonstrate its effectiveness and efficiency.

2. Extrinsic Calibration via Plane-based Structure from Motion

2.1. Problem Formulation

We consider calibrating the extrinsic parameters of $N_c \geq 2$ rigidly embedded cameras $\{C_1, C_2, \ldots, C_{N_c}\}$ with known intrinsic parameters. Since the cameras are jointly moving, there exists a constant homogeneous transformation between any two cameras. Specifically, if we choose one of the cameras as the reference camera C_r, then the homogeneous transformations from C_r to another camera C_i can be represented as

$$\begin{bmatrix} R_{ir} & \mathbf{t}_{ir} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3), \tag{1}$$

where $R_{ir} \in SO(3)$ and $\mathbf{t}_{ir} \in \mathbb{R}^3$ denote the rotation and translation, respectively. We also use $\mathbf{r}_{ir}$ to denote the Rodrigues rotation vector of R_{ir} in the experiment part. This is illustrated in Fig. 1(a). The objective of extrinsic calibration is therefore to estimate $\{R_{ir}, \mathbf{t}_{ir}\}, 1 \leq i \leq N_c$.

Our approach to extrinsic calibration is based on structure from motion (SfM), which operates by moving the camera rig into N positions and observe the scene at each viewpoint. Let X_j^k be the 3D coordinates of an object feature expressed in the coordinate system of camera C_i at the k-th frame. These coordinates are related to the coordinates of the same feature in the first frame, X_j^1, as follows:

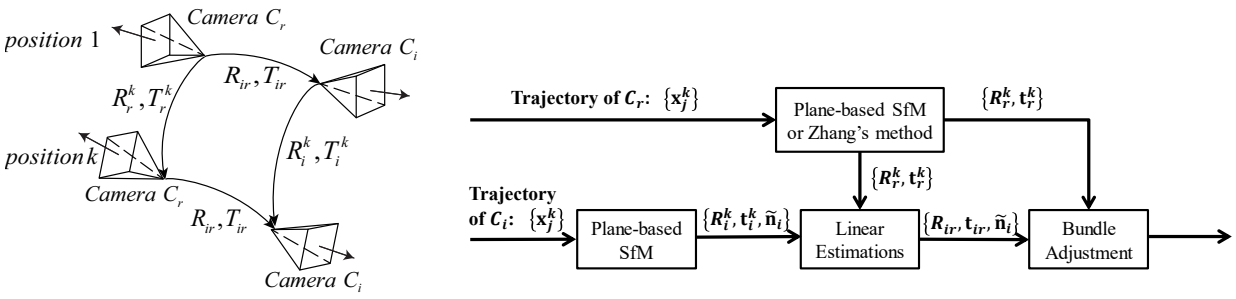
$$X_j^k = R_i^k X_j^1 + \mathbf{t}_i^k, \quad \forall 1 \leq k \leq N, \tag{2}$$

where $\{R_i^k, \mathbf{t}_i^k\}$ represents the motion of C_i at time k. Further, it is possible to show with the rigid body constraint between C_r and C_i that:

$$R_i^k R_{ir} = R_{ir} R_r^k, \tag{3}$$

$$R_i^k \mathbf{t}_{ir} + s_i \mathbf{t}_i^k = R_{ir} \mathbf{t}_r^k + \mathbf{t}_{ir}, \tag{4}$$

where s_i is a non-negative scalar that accounts for the scaling ambiguity in the magnitude of the camera translations. Therefore, if one can reliably estimate the motions of each



(a) (b)

Figure 1. Overview of our method. **(a)** Basic configuration and notation of our system. **(b)** The pipeline of our calibration method.

camera $\{R_i^k, \mathbf{t}_i^k\}_{k=1}^N$, then the homogenous transformations can be obtained by solving Eq. (3) and (4).

Fig. 1(b) provides an overview of our method. We use the dimension of C_r as the canonic dimension of the entire system. In occasions where the physical dimension of $\mathbf{t}_{ir}$ is required (e.g., robotics), we can make the reference camera observe some static scene with known metrics (e.g., a checkerboard pattern); in occasions where the exact physical dimension of $\mathbf{t}_{ir}$ is not required (e.g., 3D reconstruction), then the metrics of the scene are not needed. We compute the motion of the reference camera $\{R_r^k, \mathbf{t}_r^k\}_{k=1}^N$ using Zhang's method (for known metrics) [16] or a plane-based SfM approach (for unknown metrics) [17].

For the other cameras, it is sufficient to observe any static scene (with unknown parameters) in order to estimate their rotations R_i^k and translations $\mathbf{t}_i^k$ up to a scaling factor s_i. As shown in Fig. 1(b), the calibration process consists of computing the motion of each camera via a plane-based SfM approach, estimating the homogeneous transformations $\{R_{ir}, \mathbf{t}_{ir}\}_{i=1}^{N_c}$ by solving a linear least-squares problem, and jointly refining all variables using a global bundle adjustment.

2.2. Plane-based Motion Estimation

It is shown in [17] that both intrinsic and extrinsic parameters of a freely moving camera can be robustly and efficiently estimated by detecting and tracking one or more dominant planes in the scene, which are commonly seen in both indoor and outdoor manmade environments. For each camera C_i in our system, in this section we describe a simple plane-based method to obtain its extrinsic parameters $\{R_i^k, \mathbf{t}_i^k\}_{k=1}^N$. Then, we show how one can obtain the transformations $(R_{ir}, \mathbf{t}_{ir})$ from the camera motions.

2.2.1 Preliminaries

Given a set of N images captured by C_i, we first detect and match the feature points (i.e., SURF features) to form a set of trajectories $\mathcal{T} = \{T_j\}_{j=1}^M$ of M feature points. Each trajectory can be written as $T_j = \{\mathbf{x}_j^k\}_{k=p_j}^{q_j}$, where p_j and q_j $(1 \leq p_j \leq q_j \leq N)$ denote the starting and ending frames of the trajectory, and $\mathbf{x}_j^k \in \mathbb{P}^2$ is the coordinates of the feature point in the k-th frame. We use $\mathcal{T}^{ab} \subseteq \mathcal{T}$ to represent the set of trajectories that span the a-th and b-th frames.

In this paper, we represent a 3D plane π with respect to the first camera coordinate frame using the equation $\mathbf{n}^T X = d$, where $\mathbf{n}$ is the unit-length normal vector and $d > 0$ denotes the distance from the plane to the origin of the first camera frame. Without loss of generality, we write $\tilde{\mathbf{n}} = \mathbf{n}/d$ for simplicity. If a trajectory belongs to the 3D plane π, its coordinates in the first frame and the k-th frame are related by a planar homography $\mathbf{x}_j^k = H_i^k \mathbf{x}_j^1$, where

$$H_i^k \simeq K_i(R_i^k + \mathbf{t}_i^k \tilde{\mathbf{n}}^T)K_i^{-1}, \tag{5}$$

with the symbol $\simeq$ meaning "equality up to a scale".

2.2.2 Plane Detection and Tracking

To detect and track the dominant plane in the scene, we employ a modified version of the TRASAC algorithm [17]. Like RANSAC, this algorithm consists of multiple trials of the same procedure to select the best result. In each trial, it first picks two consecutive frames (F_{k-1}, F_k) at random. Then, it randomly choose four trajectories from $\mathcal{T}^{(k-1)k}$ to estimate the homography $H_i^{(k-1)k}$ of a putative plane model. Subsequently, each trajectory $T_j \in \mathcal{T}^{(k-1)k}$ is classified into the class of inliers $\mathcal{T}_{in}$, or the class of outliers $\mathcal{T}_{out}$, by comparing the projection error $||\mathbf{x}_j^k - H_i^{(k-1)k}\mathbf{x}_j^{k-1}||$ with a threshold ϵ. Then, the algorithm proceeds by iteratively choosing a new pair of frames adjacent to the previously chosen frames, say (F_k, F_{k+1}) (or equally (F_{k-2}, F_{k-1})), computing $H_i^{k(k+1)}$ from four randomly sampled trajectories from $\mathcal{T}^{k(k+1)} \cap \mathcal{T}_{in}$, and classifying each trajectory in $\mathcal{T}^{k(k+1)}$ into $\mathcal{T}_{in}$ and $\mathcal{T}_{out}$, until the homographies have been computed for all consecutively frames. We refer interested readers to [17] for more details about the TRASAC algorithm.

The original TRASAC algorithm assumes that if a trajectory T_j is classified as an inlier according to its fitness between two adjacent frames, it remains an inlier in all frames

it spans. While this is typically true for video sequences, a mismatch is likely to happen in other frames in our problem where the baseline between two consecutive frames is large. To exclude such kind of outliers, we post-process the output of TRASAC by moving a trajectory $T_j \in \mathcal{T}_{in}$ into $\mathcal{T}_{out}$ if

$$\sum_{k=p_j}^{q_j} \left\| \mathbf{x}_j^k - H_i^{(k-1)k} \mathbf{x}_j^{k-1} \right\|^2 > (p_j - q_j + 1)\epsilon^2. \quad (6)$$

Finally, we compute the homography from the k-th frame to the first frame as: $H_i^k = \prod_{m=1}^{k-1} H_i^{m(m+1)}$.

2.2.3 Estimation of the Extrinsic Parameters

Given the set of homographies $\{H_i^k\}_{k=1}^N$ for each camera C_i, we now discuss how to obtain the extrinsic parameters of all cameras.

First, given the intrinsic parameters K_i, we can compute $(R_i^k, \mathbf{t}_i^k, \tilde{\mathbf{n}})$ from H_i^k according to Eq. (5). It is well known that there are at most four solutions for such a decomposition [14]. With the positive depth constraint, the number of physically possible solutions can be reduced to two. Finally, by taking images at three or more general viewpoints, i.e., $N \geq 3$, we are able to identify the unique solution.

Next, given the camera motions $\{R_r^k, \mathbf{t}_r^k\}_{k=1}^N$ and $\{R_i^k, \mathbf{t}_i^k\}_{k=1}^N$ for $i = 1, 2, ..., N_c$, we can estimate the values of extrinsic parameters $\{R_{ir}, \mathbf{t}_{ir}\}$ for each pair (C_r, C_i) using Eq. (3) and (4). To estimate R_{ir}, we note that Eq. (3) can be re-written via the logarithm mapping on $SO(3)$ as [15]:

$$\mathbf{r}_i^k = R_{ir} \mathbf{r}_r^k, \quad (7)$$

where $\mathbf{r}_i^k = \log R_i^k, \mathbf{r}_r^k = \log R_r^k$. To solve for R_{ir}, we propose to minimize the Euclidean distance of the two vectors in Eq. (7):

$$\begin{aligned} \min \quad & \sum_{k=1}^{N} ||\mathbf{r}_i^k - R_{ir} \mathbf{r}_r^k||^2, \\ s.t. \quad & R_{ir}^T R_{ir} = I. \end{aligned} \quad (8)$$

The above problem has a closed-form solution

$$R_{ir} = (A_{ir}^T A_{ir})^{-\frac{1}{2}} A_{ir}^T, \quad (9)$$

where $A_{ir} = \sum_{k=1}^{N-1} \mathbf{r}_r^k (\mathbf{r}_i^k)^T$.

To estimate $\mathbf{t}_{ir}$, one can use the bisection method proposed in [9] and solve a Second Order Cone Programming feasibility problem. But such an algorithm is relatively time consuming. Since our goal is just to obtain an initial estimation of $\mathbf{t}_{ir}$, we have found that it suffices to solve the following least-squares problem from Eq. (4):

$$B_{ir} \boldsymbol{\alpha}_{ir} = \boldsymbol{\beta}_{ir} \quad (10)$$

with

$$B_{ir} = \begin{bmatrix} R_i^2 - I & \mathbf{t}_i^2 & 0 & \dots & 0 \\ R_i^3 - I & 0 & \mathbf{t}_i^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ R_i^N - I & 0 & 0 & \dots & \mathbf{t}_i^N \end{bmatrix} \in \mathbb{R}^{(3N-3)\times(N+2)}, \quad (11)$$

$$\boldsymbol{\alpha}_{ir} = \begin{bmatrix} \mathbf{t}_{ir} \\ s_i^2 \\ \vdots \\ s_i^N \end{bmatrix} \in \mathbb{R}^{N+2}, \text{ and } \boldsymbol{\beta}_{ir} = \begin{bmatrix} R_{ir}\mathbf{t}_r^2 \\ R_{ir}\mathbf{t}_r^3 \\ \vdots \\ R_{ir}\mathbf{t}_r^N \end{bmatrix} \in \mathbb{R}^{3N-3}. \quad (12)$$

Note that instead of solving for a single s_i for all frames as indicated in Eq. (4), we have found that using a different scaling factor s_i^k for each frame in Eq. (12) provides more stable estimates in the presence of noises. Finally, we simply compute $s_i = \frac{1}{N-1} \sum_{k=2}^{N} s_i^k$.

2.3. Plane-based Bundle Adjustment

The parameters obtained through the above procedure are not globally optimal. For example, the homography matrices $\{H_i^k\}_{k=1}^N$ were estimated using matched feature points between two adjacent frames, thus are not necessarily consistent with the motion of a camera observing the *same* plane across N frames. In this section, we describe a plane-based bundle adjustment method which jointly refines the extrinsic parameters of the reference camera, and homogeneous transformations from all cameras to the reference camera, and the planar scene structure.

Let $\mathcal{C} = \{C_i\}_{i=1}^{N_c}$ denote the set of all cameras. To distinguish different cameras in our objective function, we augment the notations used in previous sections as follows: $\tilde{\mathbf{n}}_i$ denotes the dominant plane seen by camera C_i, T_{ij} is the j-th trajectory observed by C_i, and (q_{ij}, p_{ij}) denote the starting and ending frames of T_{ij}. Let $\mathbf{x}_{ij}$ represent the real position (to be estimated) of the j-th feature point in the first frame of C_i[1], we propose to minimize the following geometric error function:

$$\begin{aligned} &\sum_{i \in \mathcal{C}} \sum_{T_{ij} \in \mathcal{T}_{in}} \sum_{k=p_{ij}}^{q_{ij}} \left\| \mathbf{x}_{ij}^k - K_i (R_i^k + \mathbf{t}_i^k \tilde{\mathbf{n}}_i^T) K_i^{-1} \mathbf{x}_{ij} \right\|^2 \\ &+ \sum_{i \in \mathcal{C}} \sum_{T_{ij} \in \mathcal{T}_{out}} (q_{ij} - p_{ij} + 1)\epsilon^2, \end{aligned} \quad (13)$$

where

$$R_i^k = R_{ir} R_r^k R_{ir}^T, \quad (14)$$

$$\mathbf{t}_i^k = R_{ir} \mathbf{t}_r^k + (I - R_{ir} R_r^k R_{ir}^T) \mathbf{t}_{ir}. \quad (15)$$

[1] Note that $\mathbf{x}_{ij}$ is different from $\mathbf{x}_{ij}^1$, the (possibly noisy) 2D measurement of the same quantity.

(a) (b) (c) (d)

Figure 2. Our experiment settings. **(a)** The experiment on synthetic data uses a two-camera system $\{C_r, C_t\}$ in which the camera poses are randomly generated. For calibration, we generate random feature points on a 3D plane that lie inside a rectangle observable by C_t. **(b)** A system consists of two fisheye cameras fixed in a back-to-back position. **(c)** Experimental setup of the 16-camera circular array. The calibration targets form an octagon which includes a checkerboard (facing towards the reference camera) and seven random planar patterns (facing towards the rest of the cameras). **(d)** Experimental setup of the five-camera system. We calibrate this system using street view images.

Here, ϵ is the penalty for labeling a trajectory as an outlier. Note that in this formulation we compute R_i^k and $\mathbf{t}_i^k$ using the relative poses $(R_{ir}R_r^kR_{ir}^T)$ and $(R_{ir}\mathbf{t}_r^k + (I - R_{ir}R_r^kR_{ir}^T)\mathbf{t}_{ir})$, respectively. Further, unlike the SfM systems for general 3D scenes, our plane-based bundle adjustment relates points between two frames using a homography. Therefore, it does not need to estimate the 3D coordinates of the feature points. This makes our plane-based method more robust in real world applications.

To minimize the above nonlinear objective function, we adopt an alternating scheme which iterates between estimating the structure and motion parameters and updating the trajectory labels:

- Given the classification of trajectories into $\{\mathcal{T}_{in}, \mathcal{T}_{out}\}$, we solve for $\{\mathbf{x}_{ij}\}_{j=1}^{M_i}$, R_{ir}, $\mathbf{t}_{ir}$, and $\tilde{\mathbf{n}}_i$ via the Levenberg-Marquardt method.
- Given the structure and motion parameters, we update the sets $\{\mathcal{T}_{in}, \mathcal{T}_{out}\}$ by comparing the cost of labeling a trajectory as an inlier and the cost of labeling the trajectory as an outlier:

$$T_{ij} \in \begin{cases} \mathcal{T}_{in}, & \text{if } g(i,j) < h(i,j) \\ \mathcal{T}_{out}, & \text{otherwise} \end{cases} \tag{16}$$

with

$$g(i,j) = \sum_{k=p_{ij}}^{q_{ij}} \left\| \mathbf{x}_{ij}^k - K_i(R_i^k + \mathbf{t}_i^k \tilde{\mathbf{n}}_i^T) K_i^{-1} \mathbf{x}_{ij} \right\|^2,$$

$$h(i,j) = (q_{ij} - p_{ij} + 1)\epsilon^2.$$

3. Experiments

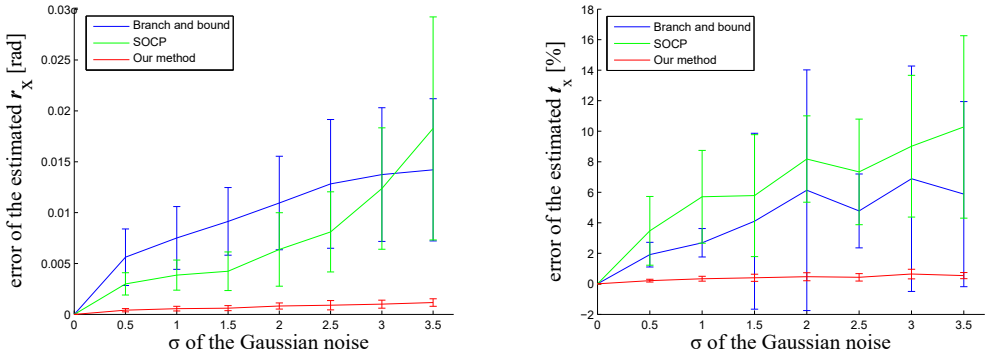
In this section, we systematically evaluate the performance of our method on both synthetic and real datasets, and compare it to the state-of-the-art methods. We first generate synthetic data to compare our method with two recent hand-eye calibration methods on calibrating a two-camera system where the cameras have non-overlapping FoVs, as shown in Fig. 2(a). We then conduct experiments on three real systems. The first system consists of a pair of Canon 5D Mark III cameras, each with a fisheye lens, fixed in a back-to-back position, as shown in Fig. 2(b). The second system is a circular camera rig composed of 16 GoPro Hero4 cameras, as shown in Fig. 2(c). We calibrate the extrinsic parameters of these two systems with some calibration targets. The third system consists of five GoPro Hero4 cameras, as shown in Fig. 2(d), and we calibrate its extrinsic parameters with images of an unknown outdoor scene. The intrinsic parameters of individual cameras were calibrated prior to the experiments using Zhang's method [16] through the MATLAB camera calibrator.

3.1. Experiments on Synthetic Data

For experiments on synthetic data, we compare our method with two recent hand-eye calibration methods, one based on Second Order Cone Programming (SOCP) [9] and the second based on branch-and-bound [8]. We use a similar experiment setting as used in the two aforementioned techniques. We construct a two-camera rig $\{C_r, C_t\}$ and set out to estimate the relative pose of C_r to C_t. Same as [9] and [8], we assume the motion of the reference camera C_r is known. We assume C_t has a resolution of 1280×1024, a focal length $f = 500$, its principle point at $(640, 512)$. We assume the camera does not have radial distortions.

Next we randomly generate the translation between C_r and C_t as $(\mathbf{r}_{tr}, \mathbf{t}_{tr})$, where $\mathbf{r}_{tr}$ is the rotation vector of R_{tr} represented through the Rodrigues formula. The configuration in Fig. 2(a) corresponds to $\mathbf{r}_{tr} = [1.4520, -0.6607, -1.1607]$ and $\mathbf{t}_{tr} = [12.2560, -225.4166, -128.9851]$. To test the robustness of all methods, we generate 100 uniformly distributed feature points inside a rectangle lying at a distance of 700 in front of C_t, as shown in Fig. 2(a). We assume that the first frame of C_t is able to observe all these points.

Finally, we randomly generate another 9 viewpoints and poses for C_t such that C_t looks approximately at the center



(a) (b)

Figure 3. Comparison of calibration methods on synthetic data with different noise levels. **(a)** The mean error and standard deviation of the estimated R_{ir}. **(b)** The mean error and standard deviation of the estimated $\mathbf{t}_{ir}$.

of the rectangle to observe the majority (but not all) feature points, and we move C_r together with C_t. Feature points are projected onto under the ground truth camera intrinsics and those lying outside the view frustum are excluded.

We test the algorithms under 8 different noise levels σ, where σ is the standard deviation of the additive white Gaussian noise on the 2D coordinates of the projected 3D feature points. To conduct the SOCP method [9], we need to first apply SfM to estimate C_t's motion: $\{R_t^k, \mathbf{t}_t^k\}_{k=1}^N$. However, a conventional SfM system fails since we use a plane as scene geometry. We instead resort to the plane-based SfM method [17] before conducting SOCP. For the branch-and-bound method [8], we directly use the source code from the authors' website.

Fig 3 compares our scheme with [9] and [8]. The standard deviation of the branch-and-bound method [8] can be rather high under large noise, i.e., its robustness degrades when the noise level is high (especially when estimating $\mathbf{t}_{ir}$). We suspect it is because, with high noise level, it may have discarded the block containing the correct solution too early. Our method, however, still gives accurate estimations in such cases, which shows that L_2-norm objective is easier to converge to a better solution than L_∞-norm under noise.

Our method also outperforms the SOCP-based method [9]. A major problem there is that it optimizes $\mathbf{t}_{ir}$ with a fixed and possibly inaccurate R_{ir}. In contrast, our method optimizes all parameters jointly via bundle adjustment. Further, our method explicitly excludes the outliers while estimating these parameters, making it less sensitive to noises.

3.2. Experiments on Real Systems

In this section, we test our method on three different real systems. In the first two experiments, we choose one of the cameras as the reference camera and face it towards a checkerboard with known scale. We detect the corners on the checkerboard in the images using [7] and estimate the reference camera's motion parameters $\{R_r^k, \mathbf{t}_r^k\}_{k=1}^N$ using Zhang's method [16]. These parameters are later jointly refined with the extrinsic parameters of the other cameras obtained by our plane-based method. In the last experiment, we do not use any calibration target with known geometry. Instead, we calibrate the 5-camera system using images of a street scene to demonstrate the wide applicability of our method. Unless explicitly stated otherwise, the parameter ϵ is set to 4 by default.

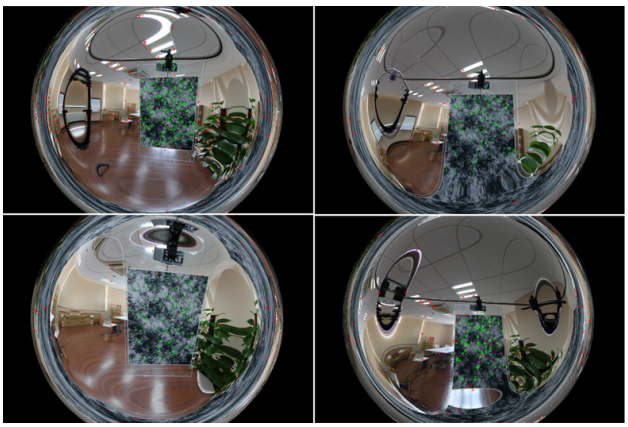


Figure 4. Example images taken by the target fish-eye camera after radial distortion correction. The green dots and red dots correspond to the inlier and outlier trajectories classified by our plane detection and tracking algorithm, respectively. For better visualization, the trajectories have been down-sampled by a factor of 3.

3.2.1 Two-Camera System, With Calibration Targets

In this experiment, we calibrate two cameras fixed in a back-to-back position (Fig. 2(b)). The cameras use Canon EF 8-15 mm fisheye lens with focal length fixed at 8 mm. We let the reference camera C_r face the checkerboard, and the target camera C_t face a random planar pattern [13]. We randomly move the camera system to take 13 pairs of

(a) (b)

Figure 5. Visualization of the results with calibration targets. **(a)** The relative positions of the 16 cameras estimated by our method. The top camera is the reference camera. **(b)** An example rectification results for two of the cameras using the estimated extrinsic parameters.

synchronized images. To remove the radial distortions, we use the Matlab camera calibrator with three-parameter radial distortion model. Fig. 4 shows some example images taken by the target camera. We have observed that it is very difficult to completely remove the radial distortion in the images. As shown in Fig. 4, even after distortion correction, only the central areas of the images are rectified. The areas near the boundary of the acquired images still exhibit substantial distortion.

In Fig. 4 we also show the inlier and outlier trajectories (i.e., matched feature points) obtained after the final refinement. It clearly shows that that our plane-based method has successfully detected the outliers caused by radial distortion. These outliers thus have no impact on the calibration results, demonstrating the robustness of our method.

The results obtained by our method are $\mathbf{r}_{tr} = [-0.0154, -3.1200, 0.0039]$, and $\mathbf{t}_{tr} = [11.1123, -0.3307, -298.1045]$. Note that the estimated $\mathbf{r}_{tr}$ is very close to 180° and $\mathbf{t}_{tr}$ is very close to the actual distance between the lenses of the two cameras, which is about 300 mm.

3.2.2 16-Camera Array, With Calibration Targets

In this experiment, we calibrate the extrinsic parameters of 16 GoPro Hero4 cameras mounted on a circular frame, as shown in Fig. 2(b). The horizontal FoV of the cameras is about 80°. The radius of the circular frame is 235 mm; the angle between neighboring cameras is 22.5°.

We choose one of the cameras as the reference and face it towards a checkerboard with known scale. We detect the corners on the checkerboard using [7] and estimate the reference camera's motion using Zhang's method [16]. We face the rest of the 15 cameras towards 7 random patterns and randomly move the camera frame (via controls on the tripod) to capture 40 sets of synchronized images. We discard the sets in which [7] fails to detect the checkerboard (e.g., due to poor illuminations) and use the remaining sets.

Fig. 5 shows the calibration results. To visually assess the quality of calibration, we use our estimated extrinsic parameters along with the cameras' intrinsic parameters to rectify neighboring pairs of images. Fig. 5(b) illustrates some sample results. In particular, the image pairs exhibits nearly pure-horizontal parallax, a sign of accurate extrinsic calibration.

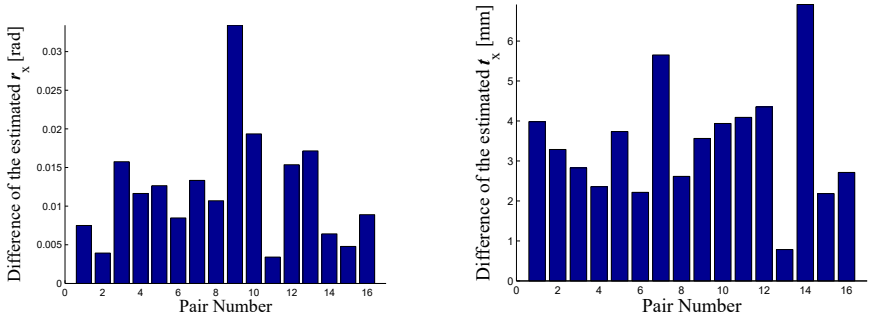
For further evaluations, we compare our results with the MATLAB stereo camera calibrator [16]. When using [16], we capture 60 pairs of images of a checkerboard to produce highly accurate rectification between every pair of adjacent cameras. The process takes about 3 hours, and it costed us 2 hours to process the data. The results also provide the extrinsic parameters between the adjacent pairs which we view as the ground truth. Next, we compare them with our estimated extrinsic parameters between adjacent pairs.

The extrinsic parameters $(R_{ir}, \mathbf{t}_{ir})$ recovered by our method refer to ones with respect to the reference camera. We can easily convert them into extrinsics between adjacent cameras, i.e., $(R_{i(i+1)}, \mathbf{t}_{i(i+1)})$. As shown in Fig. 6, the average error between our measurements vs. the ground truth across all pairs is 3.4505 mm in translation and 0.012 rad in rotation. Recall that the adjacent cameras are separate at about 92 mm. So the translation differs at about 3.75% and the view angle differs at about 3.12%. This indicates that our estimation is of high accuracy. More importantly, our method only requires 40 captures whereas we obtained the ground truth using more than 1000 captures. Our technique hence significantly improves the efficiency while maintaining high accuracy.

3.2.3 Five-Camera System, With Street Views

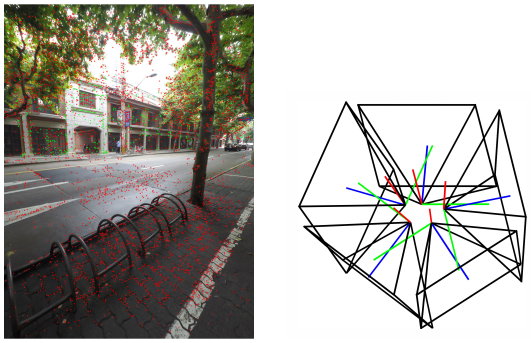
In this experiment, we calibrate the extrinsic parameters of five GoPro Hero4 cameras mounted on a circular frame, as shown in Fig. 2(d). The system is built to capture panoramic images with large vertical field of view and the minimum number of cameras. As a result, the horizontal (with respect to the circular frame) overlapping field of view is very small.

Different from previous experiments, we do not use any calibration target in the calibration. Instead, we detect the



(a) (b)

Figure 6. Comparison with the calibration result obtained by MATLAB stereo calibrator. **(a)** Difference of the estimated $R_{i(i+1)}$. **(b)** Difference of the estimated $\mathbf{t}_{i(i+1)}$.



(a) (b)

Figure 7. Calibration with street view images. **(a)** The inliers (green dots) and outliers (red dots) in one of the images taken by the reference camera. **(b)** The recovered camera poses.

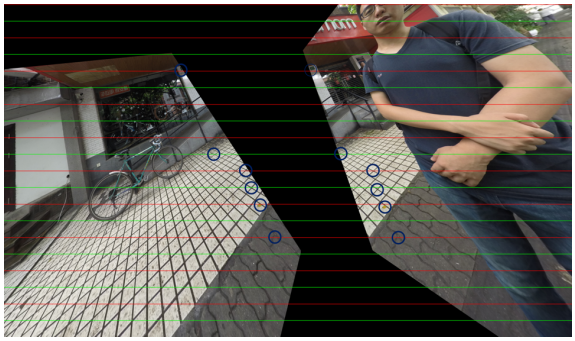


Figure 8. Rectified street view images with the estimated extrinsic parameters. We have marked some areas with blue circles for easy inspection of the vertical parallax between the two rectified images.

dominant planes in 15 street view images taken by each camera to show the generality of our algorithm. As a result, the physical dimensions of the system cannot be recovered. Note that many computer vision tasks, such as 3D reconstruction, do not require the physical dimension. Thus, we simply choose one camera as the reference camera and estimate a scaling factor for each $\mathbf{t}_{ir}$ in this system. All the cameras' poses are initialized with plane-based SfM and jointly refined with the task-specific bundle adjustment. ϵ is set to 0.7 in this experiment. One example of inlier points after bundle adjustment is shown in Fig. 7(a).

We show the estimated poses of the five cameras in Fig. 7(b). We can see from the figure that the estimated camera poses respect the geometric structure of the circular frame. To further examine the performance of our method, we also apply the calibrated extrinsic parameters to rectify a neighboring pair of images. The result is shown in Fig. 8. Although the overlapping field of view is very small in this case, we can see that the image pair still exhibits nearly pure-horizontal parallax, verifying the robustness of our algorithm.

4. Conclusion

In this paper, we proposed a robust and flexible calibration method based on planar patterns with unknown geometric parameters for calibrating multiple camera system with non-overlapping FoVs. Our method starts with the structure and motion recovery with a scene plane for each camera. Specifically, given a set of matched feature points across frames, we use a consensus method to detect the inlier trajectories corresponding to the plane, and estimate the homographies induced by the camera motion. The extrinsic camera parameters are then obtained by decomposing the homographies and solving a linear least-squares problem between camera pairs. We have further developed a task-specific bundle adjustment algorithm to jointly refine the structure and motion parameters with respect to all cameras. Experiment results on both synthetic data and real camera systems have demonstrated the effectiveness and efficiency of our method.

As future work, besides calibrating extrinsic parameters from street views, we plan to investigate reliable mechanism to obtain the intrinsic parameters by analyzing the geometric parameters of the dominant planes in natural video sequences. Such a method will benefit various applications such as stereo panoramic stitching and robotic navigation.

References

[1] N. Andreff, R. Horaud, and B. Espiau. Robot hand-eye calibration using structure-from-motion. *The International Journal of Robotics Research*, 20(3):228–248, 2001. 1

[2] H. Bay, T. Tuytelaars, and L. J. V. Gool. SURF: speeded up robust features. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, pages 404–417, 2006. 2

[3] E. A. Cansizoglu, Y. Taguchi, S. Ramalingam, and Y. Miki. Calibration of non-overlapping cameras using an external SLAM system. In *2nd International Conference on 3D Vision, 3DV 2014, Tokyo, Japan, December 8-11, 2014, Volume 1*, pages 509–516, 2014. 1

[4] G. Carrera, A. Angeli, and A. J. Davison. Slam-based automatic extrinsic calibration of a multi-camera rig. In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pages 2652–2659, 2011. 1

[5] Y. Caspi and M. Irani. Aligning non-overlapping sequences. *International Journal of Computer Vision*, 48(1):39–51, 2002. 1

[6] S. Esquivel, F. Woelk, and R. Koch. Calibration of a multi-camera rig from non-overlapping views. In *Pattern Recognition, 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007, Proceedings*, pages 82–91, 2007. 1, 2

[7] A. Geiger, F. Moosmann, O. Car, and B. Schuster. Automatic camera and range sensor calibration using a single shot. In *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*, pages 3936–3943, 2012. 6, 7

[8] J. Heller, M. Havlena, and T. Pajdla. Globally optimal hand-eye calibration using branch-and-bound. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2015. 1, 5, 6

[9] J. Heller, M. Havlena, A. Sugimoto, and T. Pajdla. Structure-from-motion based hand-eye calibration using l∞ minimization. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 3497–3503, 2011. 1, 4, 5, 6

[10] M. Knorr, W. Niehsen, and C. Stiller. Online extrinsic multi-camera calibration using ground plane induced homographies. In *2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast City, Australia, June 23-26, 2013*, pages 236–241, 2013. 1

[11] R. K. Kumar, A. Ilie, J. Frahm, and M. Pollefeys. Simple calibration of non-overlapping cameras with a mirror. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008. 1

[12] P. Lébraly, E. Royer, O. Ait-Aider, C. Deymier, and M. Dhome. Fast calibration of embedded non-overlapping cameras. In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pages 221–227, 2011. 2

[13] B. Li, L. Heng, K. Köser, and M. Pollefeys. A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013*, pages 1301–1307, 2013. 1, 2, 6

[14] Y. Ma, J. Košecká, S. Soatto, and S. S. Sastry. *An Invitation to 3-D Vision, From Images to Models*. Springer-Verlag, New York, 2004. 4

[15] F. C. Park and B. J. Martin. Robot sensor calibration: solving ax=xb on the euclidean group. *IEEE Trans. Robotics and Automation*, 10(5):717–721, 1994. 4

[16] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000. 3, 5, 6, 7

[17] Z. Zhou, H. Jin, and Y. Ma. Robust plane-based structure from motion. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 1482–1489, 2012. 2, 3, 6