Calibration of Asynchronous Camera Networks for Object Reconstruction Tasks

Amy Tabb[1] and Henry Medeiros[2]

***Abstract*— Camera network and multi-camera calibration for external parameters is a necessary step for a variety of contexts in computer vision and robotics, ranging from three-dimensional reconstruction to human activity tracking. This paper describes a method for camera network and/or multi-camera calibration suitable for specific contexts: the cameras may not all have a common field of view, or if they do, there may be some views that are 180 degrees from one another, and the network may be asynchronous. The calibration object required is one or more planar calibration patterns, rigidly attached to one another, and are distinguishable from one another, such as aruco or charuco patterns. We formulate the camera network and/or multi-camera calibration problem in this context using rigidity constraints, represented as a system of equations, and an approximate solution is found through a two-step process. Synthetic and real experiments, including scenarios of a asynchronous camera network and rotating imaging system, demonstrate the method in a variety of settings. Reconstruction accuracy error was less than 0.5 mm for all datasets. This method is suitable for new users to calibrate a camera network, and the modularity of the calibration object also allows for disassembly, shipping, and the use of this method in a variety of large and small spaces.**

I. Introduction

Camera network calibration is necessary for a variety of activities, from human activity detection and recognition, to reconstruction tasks. Internal parameters can typically be extracted by waving a calibration target in front of cameras, and then using Zhang's algorithm [25]. However, determining the external parameters, or the relationships between the cameras in the network, may be a more difficult problem, and the methods for accomplishing external camera calibration in camera networks strongly depend on characteristics of the hardware and the arrangement of the cameras. For instance, the cameras' shared field of view and level of synchronization strongly influences the ease of camera network calibration.

In this work, we provide a method for camera network calibration provided that the network meets certain conditions with respect to camera views of the patterns, to be defined in Section V-B, and the assumption that the network may not be synchronized. Our method uses calibration objects based on planar aruco or charuco patterns [8] and allows significant implementation flexibility. While we developed this approach for the application of reconstructing the shape of thin and small (i.e., 30cm $\times$ 20cm $\times$ 20cm) objects, it is suitable for synchronized networks as well. Section VI-D discusses special cases such as synchronized networks.

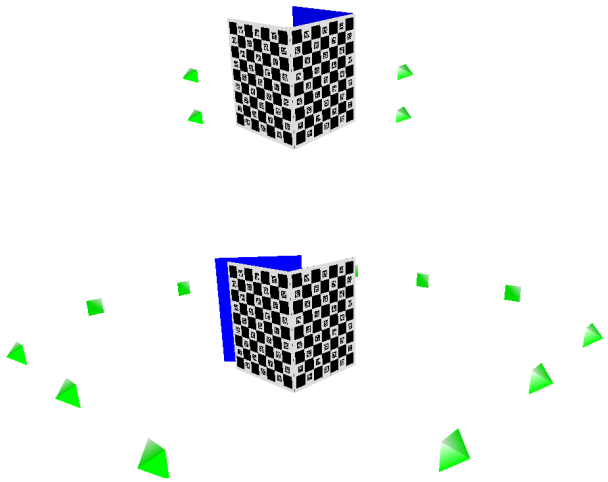

Fig. 1. **Best viewed in color** Illustrations of the two synthetic camera calibration network experiments used in this paper. A calibration rig composed of two (top) and three (bottom) planar charuco targets is moved throughout the space, and the calibration method in this paper determines the camera poses relative to the patterns without tracking. More details on these experiments are found in Section VI-B.1.

Our motivating application is a low-cost system for densely reconstructing small objects. Using a multi-view stereo paradigm, accurate camera calibration is an important element in the success of such systems [7]. The objects are from the agricultural domain, reconstructed for plant phenotyping purposes, and the method by which each object's shape is reconstructed differs ([6], [9], [15], [18], [24]). One of the experiments we will use to illustrate this paper consists of a camera network distributed across two sides of a box and pointed generally towards the center, for the reconstruction of grape rachis[1]. We require that in the future, camera networks of this type will be constructed, deconstructed, shipped, rebuilt, calibrated, and operated by collaborators in biological laboratories. Consequently, the aim of this work is that the networks may be calibrated with basic instructions, the provision of the code that accompanies the camera-ready version of this paper, and low-cost and interchangeable components.

From the above description, the use of the descriptor *camera network* is not quite accurate; camera networks usu-

[1] USDA-ARS-AFRS, Kearneysville, WV, USA amy.tabb@ars.usda.gov Mention of trade names or commercial products in this publication is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the U.S. Department of Agriculture. USDA is an equal opportunity provider and employer.

[2] Department of Electrical and Computer Engineering, Marquette University, Milwaukee, WI, USA henry.medeiros@marquette.edu

[1]Grape rachis are the stem portion of a grape cluster.

ally involve communication between nodes. However, in the literature multiple-camera systems typically refer to mobile units of cameras, such as stereo heads, multi-directional clusters of cameras (such as FLIR's Bumblebee), etc., and not to cameras in a static arrangement such as those that we consider. It is for this reason that we retain the term camera network for fixed cameras, and multiple-camera systems to cameras that may be rigidly connected, but whose base is mobile. Our calibration method may be applied to a multiple-camera system, though, and this special case is discussed in Section VI-D. Given these preliminaries, our contributions to the state-of-the-art consist of:

1) A method for the calibration of camera networks that does not depend on synchronized cameras. The method is based on the capture of a few images of a simple calibration artifact and therefore can be employed by users without a computer vision background.
2) A formulation of the calibration problem based on the iterative computation of the homogeneous transformation matrices for the individual cameras followed by the minimization of the network-wide reprojection error. This formulation does not require knowledge of the transformations between multiple rigidly attached calibration targets and is sufficiently accurate for reconstruction tasks.

II. Related Work

Camera network calibration: point targets. Synchronized camera networks, such as those used for motion capture and kinematic experiments, have long made use of a protocol of waving a wand, where illuminated LEDs are spaced at known intervals, in front of each camera. These LEDs then serve as the world coordinate system. After collecting a large number of images from each camera, structure from motion techniques are used to estimate camera parameters ([2], [3], [5], [20]).

Multi-camera calibration or asynchronous camera networks. Liu *et al.* [14] use a two-step approach to calibrating a variety of configurations, including multi-camera contexts, by using hand-eye calibration to generate an initial solution, and then minimize reprojection error. Joo *et al.* [10], working with an asynchronous camera network, used patterns projected onto white cloth to calibrate via bundle adjustment.

Robot-camera calibration. The hand-eye calibration problem, and robot-world, hand-eye calibration problem are two formulations of robot-camera calibration using camera and robot pose estimates as data. Recently, there has been interest in solving this problem optimally for reconstruction purposes. Tabb and Ahmad Yousef ([22], [23]) showed that nonlinear minimization of algebraic error, followed by minimization for reprojection error, produced reconstruction-appropriate results for multi-camera, one robot settings. Wei *et al.* [13] uses bundle adjustment to refine initial calibration estimates without a calibration target. Recently, Koide and Menegatti [12] formulated the robot-world, hand-eye calibration problem as a pose-graph optimization problem, which allows for non-pinhole camera models.

CNNs and deep learning. Convolutional neural networks (CNNs) and deep learning have been employed recently in multiple contexts to predict camera pose. For instance, [16] designed CNNs to predict relative pose in stereo images. Peretroukhin and Kelly [17], in a visual odometry context, use classical geometric and probibalistic approaches, with deep networks used as a corrector [17]. Other works focussed on appropriate loss functions for camera pose localization in the context of monocular cameras [11].

III. Hardware configuration and data acquisition

The camera networks we consider are made up of c_n cameras, which may be asynchronous. The calibration object may take many different forms.

In our implementation, we used a set of two or more planar calibration targets created with chessboard-type aruco tags ([8] and generated with OpenCV [4]), where they are referred to as charuco patterns. A three-pattern system, with a four-camera network, is shown in Figure 1. These patterns are quite convenient in that we had them printed on aluminum, which can be used outdoors and washed, and their frames can be rigidly attached to one another and then disassembled for shipment. The particular arrangement, and orientation, of the patterns is computed automatically by the algorithm; we refer to the collection of rigidly attached patterns as the calibration rig. As long as a particular calibration target's orientation can be detected, and its pattern index also detected, there is no restriction on the type of pattern used so long as the connections between individual calibration targets is rigid.

The process of data acquisition is as follows. First, multiple images are acquired per camera to allow for internal parameter calibration, or it is assumed that the cameras are already internally calibrated. Then, the user places the calibration rig in view of at least one camera. Then they indicate that this is time point 0 and acquire an image from all cameras. Then, the calibration rig is moved such that at least one or more cameras view a pattern, the user indicates that the current time is time point 1 and images are written from all of the cameras. This process if continued for the desired number of time points; minimum specifications on visibility of patterns and cameras is given in Section V-B.

IV. Camera network calibration

The camera network calibration problem consists of determining the relative homogeneous transformation matrices (HTMs) between cameras. Given the data acquisition procedure outlined in previous sections, our formulation of the problem involves three categories of HTMs: camera **C**, pattern **P** and time **T** transformations. These HTM categories are related as follows. Suppose cameras are stationary, and the pattern(s) are rigidly attached to each other, creating a calibration rig with unknown transformations between patterns. At time t_0, each camera acquires an image of the scene. Then, the calibration rig is moved. At time t_1, all cameras acquire another image of the scene. This process is repeated until time t_n. Alternative interpretations, with no change to the underlying method except for the physical

relationships of cameras to patterns, and what is stationary, versus what is moving, are discussed in Section VI-D.

Although it is important that the cameras and patterns be stationary at a particular time t, the use of 'time t_0' does not imply that the cameras are synchronized, but instead that the images be captured and labeled with the same time tag for the same position of the calibration rig. A mechanism for doing so may be implemented through a user interface that allows the user to indicate that all cameras should acquire images, assign them a common time tag, and report to the user when the capture is concluded.

Once images are captured for all t_n time steps, camera calibration of internal parameters is performed for each of the cameras independently. Individual patterns are uniquely identified through aruco or charuco tags [8]; cameras' extrinsic parameters (rotation and translation) are computed with respect to the coordinate systems defined by the patterns recognized in the image. If two (or more) patterns are recognized in the same image, that camera will have two (or more) extrinsic transformations defined for that time instant, one for each of the patterns recognized.

A. Problem Formulation

When camera c observes pattern p at time t, the HTM relating the coordinate systems of p to c can be computed using conventional extrinsic camera calibration methods. We denote this transformation as the HTM ${}^{c}\mathbf{A}_{p}^{t}$. Each HTM is composed of an orthogonal rotation matrix, a translation vector of three elements, and a row with constant terms:

$$ {}^{c}\mathbf{A}_{p}^{t} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0^{T} & 1 \end{bmatrix}. \tag{1} $$

Let ${}^{c}\mathbf{C}$ represent the world to camera transformations for camera c, ${}^{p}\mathbf{P}$ represent the calibration rig to pattern transformations p, and $\mathbf{T}^{t}$ correspond to the calibration rig transformations from the world coordinate system at time t. There is a foundational relationship (FR) between the unknown HTMs ${}^{c}\mathbf{C}$, ${}^{p}\mathbf{P}$, $\mathbf{T}^{t}$, and the known HTMs ${}^{c}\mathbf{A}_{p}^{t}$.

$$ {}^{c}\mathbf{C} = {}^{c}\mathbf{A}_{p}^{t}\,{}^{p}\mathbf{P}\mathbf{T}^{t}. \tag{2} $$

For a particular dataset, each detection of a calibration pattern results in one FR represented by equation Eq. 2. That is, let $\mathbb{C} = \{c_0, c_1, \ldots, c_n\}$ be the set of cameras, $\mathbb{T}_{p,c}$ be the set of time instants when target p is observed by camera c, and $\mathbb{P}_{t,c}$ be the set of targets observed by camera c at time t. Then, the set of foundational relationships is given by

$$ \mathbb{FR} = \left\{ \left({}^{c}\mathbf{C}, {}^{c}\mathbf{A}_{p}^{t}, {}^{p}\mathbf{P}, \mathbf{T}^{t} \right) \middle| \forall c \in \mathbb{C}, \forall t \in \mathbb{T}_{p,c}, \forall p \in \mathbb{P}_{t,c} \right\}, \tag{3} $$

where ${}^{c}\mathbf{A}_{p}^{t}$ is known and the other HTMs ${}^{c}\mathbf{C}, {}^{p}\mathbf{P}, \mathbf{T}^{t}$ are unknown.

For instance, assume camera c_0 detects pattern p_0 at times t_0 and t_1, and pattern p_1 at time t_1, and camera c_1 detects pattern p_1 at time t_1, the set of foundational relationships is given by $\mathbb{FR} = \left\{ {}^{c_0}\mathbf{A}_{p_0}^{t_0}\,{}^{p_0}\mathbf{P}\mathbf{T}^{t_0}, {}^{c_0}\mathbf{A}_{p_0}^{t_1}\,{}^{p_0}\mathbf{P}\mathbf{T}^{t_1}, {}^{c_0}\mathbf{A}_{p_1}^{t_1}\,{}^{p_1}\mathbf{P}\mathbf{T}^{t_1}, {}^{c_1}\mathbf{A}_{p_1}^{t_1}\,{}^{p_1}\mathbf{P}\mathbf{T}^{t_1}, \right\}$. Each element of $\mathbb{FR}$ corresponds to one observation for the estimation of the unknown HTMs. We describe the estimation process in Section. V.

The world coordinate system is defined by the coordinate system of a reference pattern p^* observed at a reference time t^*. Hence, ${}^{p^*}\mathbf{P} = \mathbf{I}_4$ and $\mathbf{T}^{t^*} = \mathbf{I}_4$, where $\mathbf{I}_4$ is an identity matrix of size four. We specify how p^* and t^* are chosen in Section V-C. A graphical representation of a foundational relationship is shown in Figure 2.

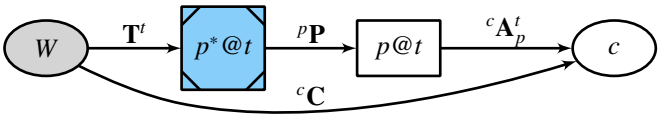

Fig. 2. Graphical representation of the foundational relationship.

V. Estimation of the unknown transformations

Our method to find the unknown transformations consists of five steps, which are summarized in Alg. 1. Each step is described in detail below.

Algorithm 1 Camera network calibration algorithm.

Input: Sets of fundamental relations $\mathbb{FR}$.

Output: Set of solutions $\mathbb{V} = \{({}^{c}\mathbf{C}, {}^{p}\mathbf{P}, \mathbf{T}^{t}) | \forall c \in \mathbb{C}, \forall t \in \mathbb{T}_{p,c}, \forall p \in \mathbb{P}_{t,c}\}$.

1: Determine intrinsic camera parameters with respect to visible patterns at all time instants.
2: Verify that the network can be calibrated using FR connectivity test.
3: Choose reference pattern p^* and time t^* and substitute the corresponding HTMs in the set $\mathbb{FR}$ where they appear.
4: Find the initial solution set $\mathbb{V}_0$ by iteratively solving individual $({}^{c}\mathbf{C}, {}^{p}\mathbf{P}, \mathbf{T}^{t})$ triples given solutions found at prior iterations.
5: Find the final solution set $\mathbb{V}$ by refining the estimated HTMs through reprojection error minimization.

A. Step 1: Intrinsic calibration of individual cameras

Step 1 is a standard component of camera calibration procedures, and will not be discussed in depth. Each pattern detection triggers the generation of one FR in Eq. 3. Note that that some images may allow the detection of more than one pattern. Also, since this step does not require knowledge of the pose of the calibration rig, it is possible to utilize images acquired as the rig is moved from position $\mathbf{T}^{t_i}$ and $\mathbf{T}^{t_{i+1}}$, if they are available.

B. Step 2: Calibration condition test

The test consists of constructing an undirected graph in which the vertices are the camera $({}^{c}\mathbf{C})$ and pattern $({}^{p}\mathbf{P})$ transformations and the edges correspond to the FRs between camera ${}^{c}\mathbf{C}$ and pattern ${}^{p}\mathbf{P}$. If the graph consists of a single connected component, then the entire network may be calibrated with this method. If the graph consists of multiple connected components, then the cameras corresponding to

each component can be calibrated with respect to each other but not with respect to the cameras in a different component.

C. Step 3: Reference pattern and time selection

The reference pattern and time are chosen such that the greatest numbers of variables can be initialized. From the list of foundational relationships, the time and pattern combination with the greatest frequency is chosen as the reference pair. That is, the reference pattern is given by

$$p^* = \arg\max_p \sum_c \left|\mathbb{T}_{p,c}\right|, \tag{4}$$

which corresponds to the pattern that has been observed the most times by the all the cameras. The reference time is given by

$$t^* = \arg\max_{t \in \mathbb{T}_{p^*,c}} \sum_c \left|\mathbb{P}_{t,c}\right|, \tag{5}$$

which is the time corresponding to the highest number of observations of target p^*. This reference pair is substituted into the list of foundational relationships, ${}^{p^*}\mathbf{P} = \mathbf{I}_4$ and $\mathbf{T}^{t^*} = \mathbf{I}_4$.

D. Step 4: Initial solution computation

We initialize the set of approximate solutions $\mathbb{V}_o = \{({}^c\mathbf{C}, {}^p\mathbf{P}, \mathbf{T}^t) | \forall c \in \mathbb{C}, \forall t \in \mathbb{T}_{p,c}, \forall p \in \mathbb{P}_{t,c}\}$ by identifying all the elements of $\mathbb{FR}$ for which $p = p^*$ and $t = t^*$ and computing the corresponding HTM ${}^c\mathbf{C}$ for all the cameras that observe the reference pair. At this stage, $|\mathbb{V}_0| \geq 1$ since at least one camera transformation can be determined from the reference pair with frequency at least one.

The solutions in $\mathbb{V}_0$ are then substituted into the corresponding elements of $\mathbb{FR}$, and the elements of $\mathbb{FR}$ for which all the transformations are known are removed from the set. Out of the remaining elements of $\mathbb{FR}$, those with only one unknown are then solved and the corresponding solutions are included in $\mathbb{V}_o$. This process is repeated until $\mathbb{FR} = \emptyset$.

1) Solving the relationship equations: Let $\mathbb{FR}^{(i)}$ be the set of elements of $\mathbb{FR}$ for which only the HTM $\mathbf{X}^{(\mathbf{i})}$ is unknown (at a given iteration, $\mathbf{X}^{(i)}$ could be either ${}^c\mathbf{C}$, ${}^p\mathbf{P}$, or $\mathbf{T}^t$). We solve Eq. 3 for the elements of $\mathbb{FR}^{(i)}$ by rearranging the terms of the relation in the form

$$\mathbf{A}\mathbf{X}^{(\mathbf{i})} = \mathbf{B}, \tag{6}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the known HTMs and $\mathbf{X}^{(\mathbf{i})}$ is the unknown transformation. If $|\mathbb{FR}^{(i)}| = 1$, we simply solve $\mathbf{X}^{(\mathbf{i})} = (\mathbf{A})^{-1}\mathbf{B}$. Otherwise, we combine all the relations in $\mathbb{FR}^{(i)}$ and solve for $\mathbf{X}^{(\mathbf{i})}$ using Shah's method [19].

2) Relationship solution order: At each iteration of the process, it may be possible to solve Eq. 3 for more than one transformation. We determine the solution order using a heuristic approach that prioritizes transformations that satisfy the highest number of constraints. That is, we select the HTM $\mathbf{X}^{(\mathbf{i})}$ that maximizes $\left|\mathbb{FR}^{(i)}\right|$. Ties are broken by choosing transformations in the order ${}^c\mathbf{C}$, ${}^p\mathbf{P}$, $\mathbf{T}^t$, and solving equations according to their indices order, if necessary.

E. Step 5: Reprojection error minimization

Once initial values for all the HTMs are estimated, they are refined by minimizing the reprojection error. Similarly to [22], [23] in the robot-world, hand-eye calibration problem, the projection matrix ${}^c\hat{\mathbf{A}}_p^t$ can be represented by

$${}^c\hat{\mathbf{A}}_p^t = {}^c\mathbf{C}\left(\mathbf{T}^t\right)^{-1} {}^p\mathbf{P}^{-1}, \tag{7}$$

and the relationship between a three-dimensional point X on a calibration pattern and the corresponding two-dimensional point x in the image is

$$x = {}^c\hat{\mathbf{A}}_p^t X. \tag{8}$$

Supposing that the detected image point that corresponds to X is $\tilde{x}$, its reprojection error is $(x - \tilde{x})^2$ or, using Eqs. 7 and 8,

$$({}^c\mathbf{C}\left(\mathbf{T}^t\right)^{-1} {}^p\mathbf{P}^{-1} X - \tilde{x})^2. \tag{9}$$

The total reconstruction error is then given by

$$re = \sum_{f \in \mathbb{FR}} \sum_{(X,x) \in \mathbb{X}_f} \left\| {}^c\mathbf{C}\left(\mathbf{T}^t\right)^{-1} ({}^p\mathbf{P})^{-1} X - x \right\|^2, \tag{10}$$

where $\mathbb{X}_f$ is the set of calibration pattern point pairs (X, x) observed in the computation of the HTM ${}^cA_p^t$ corresponding to the FR $f = \left({}^c\mathbf{C}, {}^c\mathbf{A}_p^t, {}^p\mathbf{P}, \mathbf{T}^t\right) \in \mathbb{FR}$.

We minimize Eq. 10 for all the HTMs, except those corresponding to the reference pair p^*, t^*, using the Levenberg-Marquardt algorithm implemented in the Ceres solver [1] with the elements of $\mathbb{V}_0$ as the initial solution.

VI. Experiments

The method was evaluated in synthetic as well as real-world experiments. First, we will introduce three evaluation metrics in Section VI-A, and then describe datasets and results in Sections VI-B and VI-C, respectively.

A. Evaluation

We used three metrics to evaluate the accuracy of the calibration method: algebraic error, reprojection root mean squared error, and reconstruction accuracy error.

1) Algebraic error: The algebraic error represents the fit of the estimated HTMs to their corresponding FRs. It is given by

$$ae = \frac{1}{|\mathbb{FR}|} \sum_{f \in \mathbb{FR}} \left\| {}^c\mathbf{C} - {}^c\mathbf{A}_p^t {}^p\mathbf{P}\mathbf{T}^t \right\|_F^2, \tag{11}$$

where $f = \left({}^c\mathbf{C}, {}^c\mathbf{A}_p^t, {}^p\mathbf{P}, \mathbf{T}^t\right)$ is a FR, and $\|\cdot\|_F$ denotes the Frobenius norm.

2) The Reprojection Root Mean Squared Error (rrmse): The reprojection root mean squared error is simply

$$rrmse = \sqrt{\frac{1}{N} re}, \tag{12}$$

where $N = \sum_{f \in \mathbb{FR}} \left|\mathbb{X}_f\right|$ is the total number of points observed.

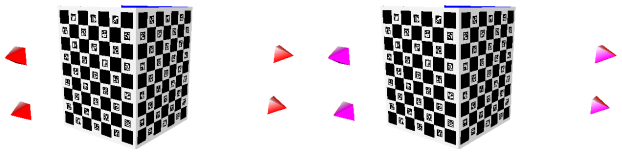

Fig. 3. Illustration of three Charuco patterns rigidly attached to each other, and four cameras observing them, using synthetic data. Left: ground truth. Right: camera positions found with our method.

3) Reconstruction Accuracy Error: The reconstruction accuracy error, *rae*, is used here in a similar way as in [23], to assess the method's ability to reconstruct the three-dimensional location of the calibration pattern points.

In [23], given detections of the same pattern point in images from different cameras at the same time, the three-dimensional point that generated the image points was estimated. The difference between the estimated and ground truth world points represents reconstruction accuracy (*rae*).

Here, *rae* is used in a slightly different way; given detections of a pattern point in images over all cameras and times, the three-dimensional point that generated those pattern points is estimated.

As before, the difference between estimated and ground truth world points represents reconstruction accuracy (*rae*). The ground truth consists of the coordinate system defined by the calibration pattern, so is known even in real settings. A more formal definition follows.

The most likely three-dimensional point X that generated the corresponding image points x can be found by solving the following minimization problem

$$\hat{\mathcal{Y}} = \underset{X}{\operatorname{argmin}} \sum_{f \in \mathbb{FR}} \left\| {}^{c}\mathbf{C}\left(\mathbf{T}^{t}\right)^{-1} \left({}^{p}\mathbf{P}\right)^{-1} X - x \right\|^{2}. \tag{13}$$

$\hat{\mathcal{Y}}$ is found for all calibration pattern points found in two or more FRs, generating the set $\mathbb{Y}$. Then, the reconstruction accuracy error (*rae*) is the average squared Euclidean distance between the estimated $\hat{\mathcal{Y}}_j$ points and corresponding calibration object points X_j.

$$rae = \frac{1}{|\mathbb{Y}|} \sum_{\hat{\mathcal{Y}}_j \in \mathbb{Y}} \left\| \hat{\mathcal{Y}}_j - X_j \right\|^{2}. \tag{14}$$

B. Datasets

1) Synthetic experiments: There are two synthetic datasets. OpenGL was used to generate images of charuco patterns from cameras with known parameters. The arrangements of the cameras are shown in Figure 1, where in the first experiment, two pairs of cameras are arranged on two perpendicular sides of a cube. The second experiment represents an arrangement more similar to that used in motion-capture experiments, where cameras are mounted on the wall around a room. For both, three charuco calibration patterns were moved rigidly within the scene.

2) Camera network: A camera network was constructed using low cost webcameras, and arranged on two sides of a metal rectangular prism, as shown in Figure 4. The

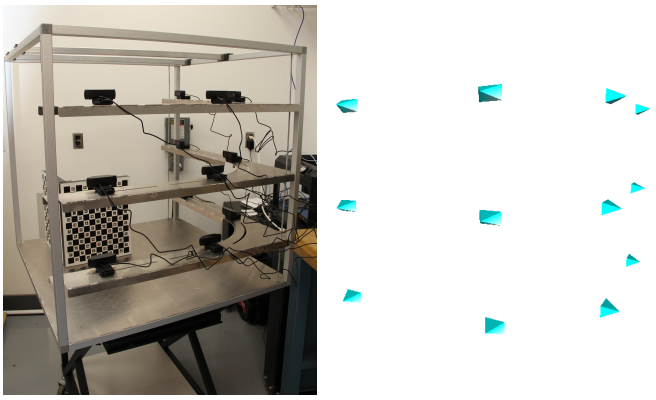

Fig. 4. **Best viewed in color.** Asynchronous camera network calibration experiment. Left: imaging system with 12 cameras, and two-pattern calibration rig. Right: computed camera locations.

calibration rig is constructed of two charuco patterns rigidly attached to each other, and data acquisition was as in Section III. Computed camera positions are shown in Figure 4, on the right.

3) Rotating object system: As mentioned previously, the method can be applied to other data acquisition contexts, such as where the goal is to reconstruct an object that is rotating and observed by one camera. In this application, the eventual goal is to phenotype the shape of fruit, such as strawberry.

In this experiment, the object was mounted on a spindle. A program triggers the spindle to turn via a stepper motor, as well as to acquire approximately 60 images from one consumer-grade DSLR camera. On the spindle are are two three-d printed cubes, which are rotated from each other by 45 degrees. A charuco pattern is mounted on each visible cube face, totalling 8 patterns. The experimental setup is shown in Figure 5, on the left side.

The calibration method from this paper is applied to this experimental design by interpreting each image acquisition of the camera as a time step. The camera is focussed between samples, so the background aruco tag image in Figure 5, coupled with exiftag information, is used to calibrate robustly for internal camera parameters.

Following the estimation of the unknown variables for the one camera, eight patterns, and approximately 60 times, virtual camera positions are generated for each image acquisition relative to the reference pattern and time. In Equation 15, ${}^{c}\mathbf{C} \in \mathbb{V}$ is the HTM representing the sole camera's pose. For all times t, virtual cameras are generated using Eq. 15.

$${}^{t}\mathbf{C} = {}^{c}\mathbf{C}\left(\mathbf{T}^{t}\right)^{-1} \tag{15}$$

These virtual camera positions are shown in Figure 5, right side as the cyan pyramids. As expected, the cameras are distributed over a circle, and the result from step 4 (right, top) is improved by minimizing reprojection error (right, middle). Using the method of Tabb [21], the shape of the object is reconstructed (right, bottom).

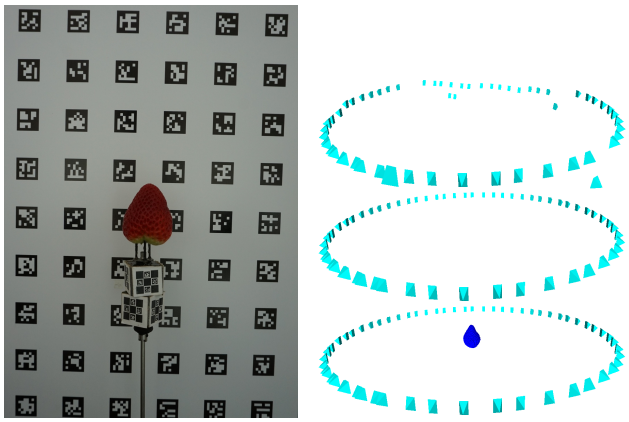

Fig. 5. **Best viewed in color** Illustration of a rotating-style data acquisition environment for shape phenotyping of fruit. Left: images from a consumer-model DLSR camera. Aruco patterns in the background are used to aid calibration for internal parameters. Right, top: visualization of camera poses, computed using the calibration method in this paper, and applied at every time step, at the conclusion of step 4 (initial solution $\mathbb{V}_0$). Right, middle: Visualization of camera poses as in prior subfigure, at the conclusion of step 5 ($\mathbb{V}$). Right, bottom: reconstruction of strawberry fruit and visualization of camera poses.

Two datasets of this type were used as experiments, one with strawberry, and another with potato.

C. Results and Discussion

The calibration method was applied to the two synthetic datasets, two rotating-style datasets, and one camera network dataset. Results in terms of the three metrics, algebraic error, reprojection root mean squared error, reconstruction accuracy error, and runtime, are shown in Table I. We implemented the method in C/C++ on a machine with a 12 core Intel Xeon(R) 2.7 GHz processor and 256 GB RAM, acquired in 2014.

Qualitatively, as shown in Figures 3 (Synthetic dataset 1), 5 (Rotating 1), and 4 (Camera Network), the estimated camera poses either visually match camera positions or where cameras are expected (in the case of rotating-style datasets). All of the experiments resulted in low *rrmse* values, though the camera network had the highest. The higher *rrmse* value of the camera network experiment versus the others is perhaps explained by that experiment's lower camera quality (i.e., webcameras), and small number of time instants versus comparably larger number of cameras.

For all of the datasets, the method produced on average, less than 1 mm^2 reconstruction accuracy error, which was surprising. For datasets with many high quality views of the calibration rig, such as the rotating-style datasets, these values are very low (< 0.0025 mm^2).

From Table I, algebraic error seems not well related to the quality of results that are of importance to reconstruction tasks. While algebraic error is used in step 4 to generate initial solutions, algebraic error may be high for for views of the calibration patterns where the estimation is not reliable. The rotating-style datasets have a high propotion of images in this category, so we hypothesize that this is why the algebraic error is so high for those datasets.

Concerning runtime, step 5, minimizing reprojection error, is the most time-consuming step of the process. Large numbers of foundational relationships heavily influences runtime. Our runtime calculations include the time to load the dataset, as well as calibrate for internal parameters and detect charuco patterns.

Impact of the number of foundational relationships. Using the camera network dataset VI-B.2, we experimented with the number of images used in the calibration. Results are shown in Table II. The minimum number of images needed to solve the calibration problem using this dataset is 4. From this dataset, as the number of images increases, the *rrmse* increases, and the *rae* decreases. This is likely because the number of constraints between HTMs increases with more images, which leads to on average, lower individual image outcomes (concerning *rrmse*), but better global outcomes in terms of *rae*. As expected, the runtime increases as the number of foundational relationships grows.

The experiment demonstrates that a 12-camera network can be calibrated with a small number of time instants, allowing its use by non-expert users.

D. Alternate data acquisition scenarios

We now discuss alternate data acquisition scenarios, beyond asynchronous camera networks, or rotating/turntable-style setups.

Consider a synchronized camera network context, such as a Vicom or Optitrak systems, where current practice is to wave wand-mounted LEDs in front of each camera. This does not take much time, but could be faster by walking through the space with two calibration patterns rigidly attached to each other. Since these systems have an extremely high frame rate, a small subset of images could be chosen to perform the calibration so as not to create an unreasonably large dataset.

Another natural context would be of a multiple camera system that is mobile, and the calibration rig is fixed. In this case, the multiple-camera system is simply moved around the calibration rig until the camera-pattern graph constraint is met (Stage 2) and the camera network problem can be solved with this method.

VII. Conclusions

We presented a method for the calibration of asynchronous camera networks, that is suitable for a range of different experimental settings. The performance of the method was demonstrated on five datasets.

Future work includes exploring ways in which it is possible to reduce runtime of step 5, the minimization of reprojection error. Possible avenues include selecting an optimal set of foundational relationships for step 5, for instance.

Other future work includes extending the calibration method to other contexts. For instance, in distributed or asynchronous camera networks, the manual triggering of data acquisition can be automated by monitoring the relative pose between the calibration patterns and the individual cameras at every frame. Once the pose differences stabilize below the

TABLE I

CALIBRATION METHOD RESULT FOR FIVE DATASETS. *ae* HAS NO UNITS, *rrmse*'S UNITS ARE PIXELS, *rae*'S UNITS ARE MM^2, AND THE UNITS FOR RUNTIME ARE SECONDS.

Dataset	$\|\mathbb{FR}\|$	cameras	patterns	times (t_n)	*ae*	*rrmse*	*rae*	runtime
Synthetic 1	18	4	3	10	0.142525	0.32111	0.0708624	46
Synthetic 2	230	12	3	40	19.7293	0.489233	0.0101121	630
Rotating set 1	162	1	8	60	6806.21	0.255644	0.00222852	338
Rotating set 2	161	1	8	61	6241.99	0.263467	0.00248751	373
Camera Box	95	12	2	10	86.1662	3.11156	0.225345	219

TABLE II

EXPERIMENT TO EXPLORE THE IMPACT OF THE VARYING THE NUMBER OF FOUNDATIONAL RELATIONSHIPS, USING THE CAMERA NETWORK DATASET. THIS DATASET HAS 12 CAMERAS, AND 2 CAMERAS. *ae* HAS NO UNITS, *rrmse*'S UNITS ARE PIXELS, *rae*'S UNITS ARE MM^2, AND THE UNITS FOR RUNTIME ARE SECONDS.

Max time	$\|\mathbb{FR}\|$	*ae*	*rrmse*	*rae*	runtime
4	37	122.164	2.85215	1.44212	94
5	49	84.235	2.60504	0.518698	114
6	58	92.5878	2.84392	0.671664	153
7	58	92.5878	2.84392	0.671664	136
8	74	69.1624	2.46442	0.335494	170
9	85	96.972	2.77998	0.280364	195
10	95	86.1662	3.11156	0.225345	219

expected pose estimation error, the object can be considered stationary, triggering image capture across all the cameras.

ACKNOWLEDGMENTS

We gratefully acknowledge the use of the rotating datasets from Mitchell Feldmann in Steven J. Knapp's lab; their work is supported in part by University of California and grants to S.J.K. from the USDA National Institute of Food and Agriculture Specialty Crops Research Initiative (#2017-51181-26833) and California Strawberry Commission.

REFERENCES

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

[2] P. Baker and Y. Aloimonos. Complete calibration of a multi-camera network. pages 134–141. IEEE Comput. Soc, 2000.

[3] N. A. Borghese, P. Cerveri, and P. Rigiroli. A fast method for calibrating video-based motion analysers using only a rigid bar. *Medical & Biological Engineering & Computing*, 39(1):76–81, Jan. 2001.

[4] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[5] L. Chiari, U. D. Croce, A. Leardini, and A. Cappozzo. Human movement analysis using stereophotogrammetry: Part 2: Instrumental errors. *Gait & Posture*, 21(2):197–211, Feb. 2005.

[6] W. Dong and V. Isler. Tree Morphology for Phenotyping from Semantics-Based Mapping in Orchard Environments. *arXiv:1804.05905 [cs]*, Apr. 2018. arXiv: 1804.05905.

[7] Y. Furukawa and C. Hernández. Multi-View Stereo: A Tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

[8] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, June 2014.

[9] F. Hui, J. Zhu, P. Hu, L. Meng, B. Zhu, Y. Guo, B. Li, and Y. Ma. Image-based dynamic quantification and high-accuracy 3d evaluation of canopy structure of plant populations. *Annals of Botany*, 121(5):1079–1088, Apr. 2018.

[10] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. S. Godisart, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. A. Sheikh. Panoptic Studio: A Massively Multiview System for Social Interaction Capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.

[11] A. Kendall and R. Cipolla. Geometric Loss Functions for Camera Pose Regression with Deep Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6555–6564, July 2017.

[12] K. Koide and E. Menegatti. General Hand–Eye Calibration Based on Reprojection Error Minimization. *IEEE Robotics and Automation Letters*, 4(2):1021–1028, Apr. 2019.

[13] W. Li, M. Dong, N. Lu, X. Lou, and P. Sun. Simultaneous robot–world and hand–eye calibration without a calibration object. *Sensors*, 18(11), 2018.

[14] A. Liu, S. Marschner, and N. Snavely. Caliber: Camera Localization and Calibration Using Rigidity Constraints. *International Journal of Computer Vision*, 118(1):1–21, May 2016.

[15] S. Liu, L. M. Acosta-Gamboa, X. Huang, and A. Lorence. Novel Low Cost 3d Surface Model Reconstruction System for Plant Phenotyping. *Journal of Imaging*, 3(3):39, Sept. 2017.

[16] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Relative Camera Pose Estimation Using Convolutional Neural Networks. In J. Blanc-Talon, R. Penne, W. Philips, D. Popescu, and P. Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, Lecture Notes in Computer Science, pages 675–687. Springer International Publishing, 2017.

[17] V. Peretroukhin and J. Kelly. DPC-Net: Deep Pose Correction for Visual Localization. *IEEE Robotics and Automation Letters*, 3(3):2424–2431, July 2018. arXiv: 1709.03128.

[18] H. Scharr, C. Briese, P. Embgenbroich, A. Fischbach, F. Fiorani, and M. Müller-Linow. Fast High Resolution Volume Carving for 3d Plant Shoot Reconstruction. *Frontiers in Plant Science*, 8, 2017.

[19] M. Shah. Comparing two sets of corresponding six degree of freedom data. *Computer Vision and Image Understanding*, 115(10):1355–1362, Oct. 2011.

[20] R. Summan, S. G. Pierce, C. N. Macleod, G. Dobie, T. Gears, W. Lester, P. Pritchett, and P. Smyth. Spatial calibration of large volume photogrammetry based metrology systems. *Measurement*, 68:189–200, May 2015.

[21] A. Tabb. Shape from Silhouette Probability Maps: Reconstruction of Thin Objects in the Presence of Silhouette Extraction and Calibration Error. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 161–168, June 2013.

[22] A. Tabb and K. M. Ahmad Yousef. Parameterizations for reducing camera reprojection error for robot-world hand-eye calibration. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3030–3037, Sept. 2015.

[23] A. Tabb and K. M. Ahmad Yousef. Solving the robot-world hand-eye(s) calibration problem with iterative methods. *Machine Vision and Applications*, 28(5):569–590, Aug. 2017.

[24] A. Tabb and H. Medeiros. A robotic vision system to measure tree traits. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6005–6012, Sept. 2017.

[25] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov. 2000.