Joint calibration of multiple sensors

Quoc V. Le and Andrew Y. Ng

***Abstract*— Many calibration methods calibrate a pair of sensors at a time. For robotic systems with many sensors, they are often time-consuming to use, and can also lead to inaccurate results. In this paper, we combine a number of ideas in the literature to derive a unified framework that jointly calibrates many sensors a large system. Key to our approach are (i) grouping sensors to produce 3D data, thereby providing a unifying formalism that allows us to jointly calibrate all of the groups at the same, (ii) using a variety of geometric constraints to perform the calibration, and (iii) sharing sensors between groups to increase robustness. We show that this gives a simple method that is easily applicable to calibrating large systems. Our experiments show that this method not only reduces calibration error, but also requires less human time.**

I. Introduction

Calibration remains a challenging and time-consuming task in robotics despite being a prerequisite for the success of many applications, such as manipulation. The problem of accurate calibration is especially pronounced on robots equipped with multiple sensors such as the STanford AI Robot (STAIR).

Researchers have proposed many techniques for calibrating specific types of sensors or pairs of sensors. Typical examples include one-camera, camera-to-camera [1] and laser-to-camera calibration [2]. There is, however, a lack of methods that jointly (simultaneously) calibrate a large system consisting of multiple sensors. A standard approach is to divide the system into many pairs of sensors and calibrate each pair at a time. In this process, one needs to use different algorithms for different pairs. Consequently, system calibration becomes very difficult, time-consuming, and often also inaccurate.

The main contribution of our paper is to combine various ideas in the literature to derive a *unified framework* that jointly calibrates a large system. Key to our approach is the use of one objective function for the entire system. The framework comprises two main ideas. First, we group different sensors on the robot to form systems that output 3D data (distances to visible objects). For example, two cameras might be grouped together to form a stereo vision system. Then we use geometric constraints (such as distance preservation, collinearity, coplanarity) to form an objective function for the entire system. We call our approach *joint calibration*.

Interestingly, we recommend *sharing* sensors between groups. That is, if there are many possible combinations of sensors that can produce 3D data, we recommend having as many pairs as possible in the objective function, so that a single sensor may participate in many different groups. This redundancy in the objective adds robustness to the calibration result, and our experiments show that it leads to reduced error.

An advantage of having groups that output 3D data is that the same calibration objective can now be used for the "extrinsic" calibration problem of calibrating the groups relative to each other, regardless of the type of sensor. More specifically, extrinsic calibration of these systems can be done via a closed form solution (Horn's algorithm [3], which requires solving for the eigenvectors of a symmetric 4x4 matrix) or via numerical optimization methods. We also combine this with an objective for "intrinsic" calibration (of the sensor parameters within each group), so that intrinsic and extrinsic calibration of all the sensors is performed simultaneously.

Our approach can be applied to many practical robotic systems because most sensors in such systems either already output 3D data or can be grouped to have this feature: examples include active triangulation sensors, stereo cameras, range finders and especially robot arms.[1]

Our experiments illustrate the advantages of the joint calibration approach. Compared to standard techniques, our method is more accurate and requires less human supervision. On our robot platform (STAIR), we are able to jointly calibrate many sensors to a 5 DOF robot arm with an average error of less than two millimeters. Using this accurate calibration, our robot is able to perform manipulation tasks such as turning knobs, pressing small elevator buttons, opening drawers and picking up small objects.

II. Previous Work

The field of sensor calibration has a long history. Camera calibration is a well studied subject [4], [5], [6]. Jointly calibrating the intrinsic and extrinsic parameters of cameras has been studied as well; for example, Zhang [5] proposed jointly estimating the intrinsics and extrinsics using one objective function. There is also software for camera-to-camera calibration, such as the well-known Caltech calibration toolbox [1], that implements this idea. Calibrating *multiple* cameras simultaneously has recently gained some interest. For example, using silhouette geometry constraints, [7] suggest a bundle optimization method that allows a joint optimization of many cameras.

The problem of calibrating different types of sensors together has received less attention, but one example is [8],

Quoc V. Le and Andrew Y. Ng are with the Computer Science Department, Stanford University quocle@cs.stanford.edu, ang@cs.stanford.edu

[1]For robot arms, intrinsic calibration corresponds to finding the kinematic parameters.

which proposed an algorithm to calibrate a camera to a range finder using plane constraints. A similar algorithm is implemented in the CMU laser-to-camera calibration toolbox [2]. Their techniques are simple yet rather specific to the properties of the sensors, and there are no known extensions to other types of sensors.

There are also works concentrating on calibrating the kinematic parameters of robotic manipulators [9], [10], [11], [12], [13]. The goal is to obtain very high precision for (primarily) industrial robots. Zhang and Roth [14] divide robot calibration into four sub-tasks: kinematic modelling, pose measurements, kinematic identification and kinematic compensation. The calibration technique in our work focuses on pose measurements and kinematic identification. In [9], authors describe techniques for identifying the static and dynamic parameters of a manipulator. Some approaches use the aid of extra sensors such as cameras [13] to calibrate the kinematic parameters. In [15], the authors show a statistical method for calibrating the odometry parameters of a mobile robot. Recently, methods to learn the kinematic parameters and models for legged robots [16] or manipulators [17] have been proposed.

Most prior work in sensor-to-arm calibration such as the early articles by Tsai et al. [18], [19], concentrates primarily on eye-in-hand calibration, where a camera is rigidly attached to a robot arm. Also for this configuration, Horaud and Dornaika [20] propose a numerical method while Daniilidis [21] uses dual quaternions. Simultaneously calibrating the hand and the camera is the focus of [22], [23]. While eye-in-hand calibration is beyond the scope of our work, the ideas of numerical optimization (e.g., [20]) and simultaneous calibration (e.g., [22]) are also used in our approach.

In general, most state-of-the-art calibration techniques are hardware-specific and do not generalize well to other types of sensors. As a result, calibrating a multi-sensor system requires both understanding the low-level details of each sensor and applying a variety of specialized algorithms. This approach is prohibitively expensive and complex for large systems. We present a unified framework that incorporates some of the ideas described above, and develop a general technique for joint calibration of multiple sensors.

III. Summary of joint calibration

In this section we present a summary of our approach. Consider the problem of calibrating the intrinsic and extrinsic parameters of a robot system which contains two cameras, a laser projector[2] and a robot arm.

A standard procedure for calibrating such a system is shown in Fig. 1. For example, we might first calibrate the pair of cameras to each other, then a camera to the laser projector, and so on. Sensors are grouped into smaller systems and each system is calibrated separately. Although there exist algorithms for each of these groups, the principles behind the groupings are unclear. Also, note that at each calibration step we have to collect different datasets and use different algorithms. Finally, this procedure can often lead to inaccurate calibrations.[3]

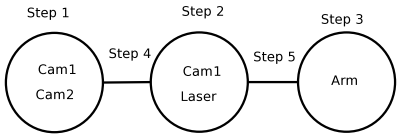

Fig. 1. Standard calibration requires many steps to calibrate an entire system.

In contrast, our approach to joint calibration is summarized in Fig. 2. The groups are formed using a unifying principle: each group is created to *output 3D data*.[4] We make use of the fact that a sensor can appear in different groups to have redundancy. Then we calibrate the entire system in one optimization.

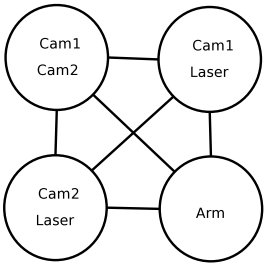

Fig. 2. Joint calibration, in which all calibration parameters are simultaneously solved for. Note the sharing and redundancy enjoyed by this method: One sensor can appear in many groups (shown by circles), and multiple links (each representing a coordinate transformation) connect the different groups. Each group is capable of outputting 3D data.

In detail, we use geometric constraints (distance preservation, collinearity, coplanarity) for intrinsic calibration within each group. Extrinsic calibration between groups can be done via Horn's algorithm [3] or Levenberg-Marquardt optimization. More importantly, although intrinsic and extrinsic calibration could have been done separately, we argue instead that intrinsic and extrinsic calibration should be combined and performed simultaneously. In our approach, this is done by posing a single calibration optimization objective, that is minimized through numerical optimization.

IV. Extrinsic and Intrinsic Parameters

A. Extrinsic parameters

The extrinsic parameters of two coordinate systems consist of a rotation matrix and a translation vector between the frames. More specifically, a 3D coordinate transformation takes a point in one frame and gives the coordinate of the point in another frame. Associated with this transformation

[2]This projector can only send a laser beam and does not measure the distance directly.

[3]For example, suppose sensor 1 (or group 1) is calibrated to sensor 2, which is calibrated to sensor 3, and so on in a long chain. If each calibration step introduces even a small amount of error, then the error in the transformation between sensor 1 and sensor n may become large, since the errors along the chain would accumulate.

[4]Here, we assume that grouping Cam1 and Laser projector will result in an active triangulation sensor (as described in Section VIII-B).

are a rotation matrix and a translation vector, which we denote by $\mathbf{R}$ and $\mathbf{t}$. Assume the coordinate of a point p is $[x, y, z]^T$ in frame A; its coordinates in frame B is then given by

$$\mathbf{R}[x, y, z]^T + \mathbf{t} \tag{1}$$

The rotation matrix also satisfies and orthonormal constraint:

$$\mathbf{R}^T\mathbf{R} = \mathbf{I} \tag{2}$$

B. Intrinsic models and parameters

We define a 3D system as a sensor or a group of sensors that gives us 3D data with known depth scales. Informally, a 3D system can measure 3D coordinates of objects in front of it. With such systems we can define an intrinsic model that takes intrinsic parameters and data and returns a point in 3D

$$\mathcal{I}(\alpha, u) \in \mathbb{R}^3 \tag{3}$$

where u is a calibration datum (e.g., corresponding pixels in two images), and α are the intrinsic model parameters. Here α may be sensor dependent; for example, for a stereo vision system, α would include the standard camera parameters (such as focal length, distortion, etc.) as well as the transformation between the two cameras.

Note that our concept of intrinsic parameters is different from that in the camera calibration literature. In that literature, a camera's intrinsic parameters include focal length, distortion, principal point, skewness, etc. In contrast, in our setting the intrinsic parameters of a 3D system may also include the rotation and translation between different members of a group.

In our definition, a single camera is *not* a 3D system because it does not produce known scaled depth data. In contrast, a 3D range finder or a stereo camera are 3D systems.

V. Extrinsic calibration of 3D systems

This task requires us to find rotation and translation parameters between systems, or informally, to find locations and orientations of different frames relative to each others. To simplify notations, we begin by considering the task of calibrating two 3D systems.

One way to find rotation and translation from one frame to another frame given corresponding 3D data is via a closed form solution given by Horn's algorithm [3].

Horn's algorithm is elegant, but can be used only to find extrinsic parameters given point constraints. In order to also address intrinsinc calibration, and to incorporate a broader set of constraints than point constraints (such as line and plane constraints, described later), we will instead use a numerical optimization method. In detail, assume a point $p^{(A)} = \mathcal{I}_A(\alpha_A, u^{(A)})$ in frame A and that same point in frame B is $p^{(B)} = \mathcal{I}_B(\alpha_B, u^{(B)})$. Now, the coordinate of $p^{(A)}$ in frame B under the coordinate transformation is $\mathbf{R}p^{(A)} + \mathbf{t}$.

Perfect extrinsic parameters will make $\mathbf{R}p^{(A)} + \mathbf{t} = p^{(B)}$. As data may be noisy, we can model the distance $d = \|\mathbf{R}p^{(A)} + \mathbf{t} - p^{(B)}\|$ as a normal random variable $d \sim \mathcal{N}(0, \lambda^2)$. Assume that we can collect many corresponding points $\{u_i^{(A)}, u_i^{(B)}\}_{i=1}^n$, maximum likelihood estimation requires us to minimize

$$Extr(\mathbf{R}, \mathbf{t}) = \sum_i \|\mathbf{R}\mathcal{I}(\alpha_A, u_i^A) + \mathbf{t} - \mathcal{I}(\alpha_B, u_i^B)\|^2 \tag{4}$$

subject to the orthonormal constraint (Eq. 2). Calibrating multiple sensors can be done in the same fashion by adding more terms to the objective function.

VI. Intrinsic calibration of 3D systems

This task requires us to find intrinsic parameters of 3D systems. Informally, this means we have to determine the internal parameters of a 3D system such that it obeys geometric rules. For example, if the world has a set of points belonging to a line, then from the perspective of the sensor, the points should also form a line.

The set of constraints we can use for intrinsic calibration of a 3D system are distance preservation, collinearity, coplanarity.[5] In our framework, these constraints are enforced by likelihood estimation.

A. Distance preservation constraints

Suppose we can collect calibration data u_i and u_j with known distance $\bar{d}_{ij}$ in 3D. Intrinsic models of a 3D system give us two points in 3D, p_i and p_j. The distance d_{ij} between p_i and p_j is represented by a random normal variable $d_{ij} \sim \mathcal{N}(\bar{d}_{ij}, \sigma^2)$, or more explicitly

$$p(d_{ij}) = \frac{1}{Z}\exp(-\frac{\|d_{ij} - \bar{d}_{ij}\|^2}{\sigma^2}) \tag{5}$$

An example of distance preservation constraints arise in the case of a stereo intrinsic model of two cameras (see Fig. 3). Here $u_i = \{u_i^{(1)}, u_i^{(2)}\}$ which are the corresponding pixels of two cameras' images. Using u_i, this system can construct a point in 3D p_i. Likewise, u_j gives p_j . We can use the ground truth distance $\bar{d}_{ij}$ of these two points as a distance preservation constraint.

B. Collinearity and Coplanarity constraints

Not only distance preservation can be employed as calibration constraints, the knowledge of some points belong to a plane or a line can also be used to help calibration.

Suppose some of our calibration data belong to a line or a plane, we can model the distance d from the points to the line/plane as a normal random variable $d \sim \mathcal{N}(0, \sigma^2)$

$$p(d) = \frac{1}{Z}\exp(-\frac{\|d\|^2}{\sigma^2}) \tag{6}$$

In detail, distances d's can be obtained by fitting a line or a plane to a set of points (via SVD - singular value decomposition [24]) and computing the distances of the points to the line or the plane.

[5]Note, that these sets of constraints can also be used for extrinsic calibration.

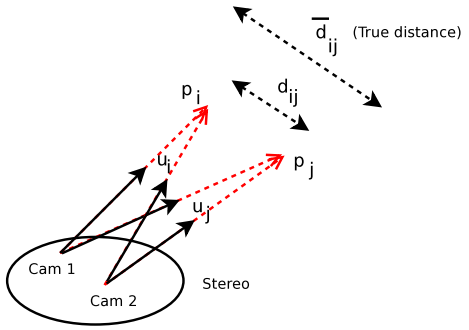

Fig. 3. An example of distance preservation constraints. Cameras $Cam1$ and $Cam2$ use a datum u_i (corresponding pixels in two cameras) to find a point $p_i \in \mathbb{R}^3$ and use a datum u_j to find a point $p_j \in \mathbb{R}^3$. We know ground truth distance between them $\bar{d}_{ij}$ (size of checkerboard square), current calibration parameters give hypothetical distance d_{ij}. By forcing d_{ij} to be close to $\bar{d}_{ij}$, we get good estimate of calibration parameters.

C. Generating constraints

Distance preservation constraints can be generated by a standard "checkerboard procedure." For example, in the case of stereo cameras, we can collect images of a checkerboard with *known* square sizes and use a corner detector algorithm to find the corners (e.g., the algorithm in [25]).

We can generate collinearity and coplanarity constraints by using flat (planar) surfaces as calibration targets. For example, flat walls, desks, checkerboards give plane constraints; whereas intersections between them give line constraints.

Coplanarity constraints can be generated more easily and quickly than distance preservation constraints because corner detection can sometimes be imprecise. There is, however, a trade-off between distance preservation and coplanarity constraints because artificial flat surfaces may not be exactly planar. Also, one can notice that line and plane constraints suffer from the scale problem: we can scale all points down to one single point and get a much higher likelihood.[6]

D. Combining all constraints

Combining likelihood of the constraints in Eq. 5 and 6, maximum likelihood estimation requires us to minimize

$$\begin{aligned} Intr(\alpha) = &\sum_i (\|\mathcal{I}(\alpha, u_i) - \mathcal{I}(\alpha, u_j)\| - \bar{d}_{ij})^2 \\ &+ \sigma_1^2 \sum_{\mathcal{L}} \sum_{k \in \mathcal{L}} d(\mathcal{I}(\alpha, u_k), \mathcal{L})^2 \\ &+ \sigma_2^2 \sum_{\mathcal{P}} \sum_{l \in \mathcal{P}} d(\mathcal{I}(\alpha, u_l), \mathcal{P})^2 \end{aligned} \quad (7)$$

where $d(x, \mathcal{L})$ is the distance from a point x to a line $\mathcal{L}$ and $d(x, \mathcal{P})$ is the distance from a point x to a plane $\mathcal{P}$.

VII. Joint intrinsic and extrinsic calibration of 3D systems

Assume we have two 3D systems and want to calibrate their intrinsic and extrinsic parameters. Using the above ideas, we first calibrate the intrinsics of each system, and *then* calibrate the extrinsics between them. However, in a similar approach to Zhang's idea [5], we should combine the two steps and jointly calibrate all parameters at once.

Again, using the maximum likelihood estimation, we need to minimize

$$\begin{aligned} ExtrIntr(\mathbf{R}, \mathbf{t}, \alpha_A, \alpha_B) = &Intr(\alpha_A) + Intr(\alpha_B) \\ &+ \gamma Extr(\mathbf{R}, \mathbf{t}, \alpha_A, \alpha_B) \end{aligned} \quad (8)$$

where the optimization variables in the above are $\alpha_A, \alpha_B, \mathbf{R}, \mathbf{t}$ and γ is the tradeoff between the intrinsic and extrinsic objectives. The difference of this approach to the two-step incremental approach is that we are going to estimate intrinsic and extrinsic parameters jointly.

VIII. Some 3D systems

Although a camera is not a 3D system, there are many 3D systems in a standard robot system. Stereo cameras, active triangulation sensors, range finder sensors and robotic arms are examples of such 3D systems. This section explains these 3D systems and their intrinsic models in more details.

A. Stereo systems

A stereo system contains two or more cameras. With stereo systems, depth is found by triangulation [26], [27]. Given corresponding pixels in two images, when there is no occlusion, we can find a unique 3D point. For simplicity and clarity, in this section we consider stereo systems with two cameras. We define the following *Stereo* function

$$Stereo(\{\mathbf{R}, \mathbf{t}, \alpha_{cam}^{(1)}, \alpha_{cam}^{(2)}\}, \{u^{(1)}, u^{(2)}\}) = [x, y, z]^T \quad (9)$$

where $\mathbf{R}, \mathbf{t}$ are the rotation and translation parameters between the two cameras, $\alpha_{cam}^{(1)}, \alpha_{cam}^{(2)}$ are parameters of the two cameras. The parameters $\mathbf{R}, \mathbf{t}, \alpha_{cam}^{(1)}, \alpha_{cam}^{(2)}$ form the set of intrinsic parameters of a stereo system, i.e $\alpha_S = \{\mathbf{R}, \mathbf{t}, \alpha_{cam}^{(1)}, \alpha_{cam}^{(2)}\}$. This $Stereo$ function takes pixel coordinates $u^{(1)}, u^{(2)}$ in two images of the two cameras and give us a point in 3D. Calibration data are $u^{(1)}$ and $u^{(2)}$.

B. Active triangulation systems

A typical active triangulation system contains a laser projector (scanner) and a camera [28], [29]. The laser projector sends a laser pattern and the camera captures a series of images. A motor on the laser projector records offset angles. An example of active triangulation system with a line pattern is described in [28].

An active triangulation system finds 3D location of image pixels. We define the following function

$$ActiveTriangulation(\{\mathbf{R}, \mathbf{t}, \alpha_{cam}\}, \{\beta, u\}) = [x, y, z]^T \quad (10)$$

that takes rotation $\mathbf{R}$, translation $\mathbf{t}$, camera intrinsic α_{cam}, pixel coordinates u and laser offset angle β and gives us a point in 3D. Calibration data are β, u.

The parameters $\mathbf{R}, \mathbf{t}, \alpha_{cam}$ form the 'intrinsic' parameters of an active triangulation system, i.e. $\alpha_{ActiveTriangulation} = \{\mathbf{R}, \mathbf{t}, \alpha_{cam}\}$.

[6] To alleviate this, we must at least have one distance preservation constraint.

C. Robot arms

Given a robot arm, the job of forward kinematics is to compute the position and orientation of the end effector [30] given the link lengths, encoder offsets, joint angles and joint angle offsets. For calibration, we only consider the 3D position of the end effector

$$ForwardKinematics(\{\mathbf{l}, \beta\}, \mathbf{e}) = [x, y, z]^T \quad (11)$$

where $\mathbf{e}$ is the encoder readings, β is the joint angle offsets and encoder offsets, and $\mathbf{l}$ is the link lengths.[7]

Here, we treat the link lengths and angle offsets, encoder offsets $\{\mathbf{l}, \beta\}$ as intrinsic parameters and encoder readings $\mathbf{e}$ as calibration data.

An advantage of using encoder readings as calibration data is that encoder readings are usually very accurate while link lengths or angle offsets are harder to measure. For example, if the last link of the arm is modified, it is hard to accurately measure its length.

D. Range finders

The intrinsics of off-the-shelf range finders, such as a laser scanner (SICK), are already well calibrated internally by the manufacturers. We can assume that these sensors do not have any intrinsic parameters that need further calibration. Thus, the range finder operation can be defined as follows

$$RangeFinder(u) = [x, y, z]^T \quad (12)$$

where calibration data u can be a 'pixel' location in the depth map image.

IX. Calibration of non-3D systems

There are many sensors that are not 3D, for example, cameras. There are two possible solutions for calibrating them.

One immediate solution is to directly apply our joint calibration method to non-3D systems at the cost of dealing with sensor-dependent problems. For example, we can calibrate a camera to a camera using the extrinsic and intrinsic calibration ideas described above. The only difference is that we have to treat the scale factors as optimization variables. This, however, violates our framework because the scale factor is a sensor-dependent variable.

A better solution is to build 3D systems out of non 3D systems. For example, we can group two cameras together and have a 3D system. Another example is that we can group a laser projector and a camera to have a 3D active triangulation system.

Grouping is advantageous because it simplifies calibration. For example, it is very challenging to calibrate a camera with a laser range finder [8], [2]. This is because the camera gives 3D rays whereas a laser range finder gives 3D points. However, if we find another camera and group two cameras together, calibrating a laser range finder vs. a stereo system can be easier.

[7]Here, we only consider static parameters.

X. Optimization

Although the above optimization objectives are nonconvex, they have a small number of parameters. They are also in the form of sum of squares. We can thus use the Levenberg-Marquardt algorithm to find a local minimum [31], [32], [33]. Software packages such as MATLAB has an implementation of this algorithm.

Levenberg-Marquardt is fast but can converge to poor local minima. To alleviate this problem, we use random restart.[8] Using random restart, we usually obtain better local minima.

In our experiments, we used numerical differentiation [33]. This is because we treat each component black-box and it is hard to take analytical derivatives. Numerical differentiation is implemented in MATLAB.

Finally, there are many ways to enforce the orthonormal constraints. In our experiments, we added $\nu\|\mathbf{R}^T\mathbf{R} - \mathbf{I}\|_2^2$ to the objective function with large value of ν. This usually works very well for us.

XI. Experiments

We perform calibration experiments with our robot (see Fig. 4). Our robot has one 5 DOF Katana arm, two visible light cameras, one Pan-Tilt-Zoom camera[9] and one laser projector. Our goal is to calibrate all of these devices against the arm.

In the next sections, we will first describe the calibration of a 3D system (camera and laser projector) and then describe the joint calibration of the whole system.

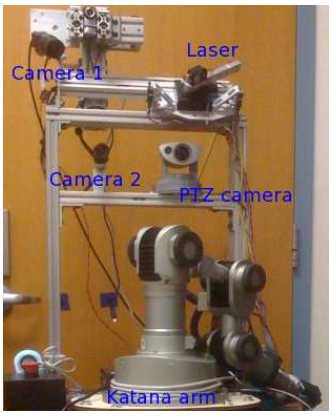

Fig. 4. Our robot platform with all sensors and the arm used in the experiment.

In the experiments, optimization variables are initialized by simple guesses. For example, the translation vector between two cameras can be measured by a ruler, the rotation is set to the identity matrix.

[8]Every time the algorithm converges we restart the algorithm by adding some perturbations to the best-so-far parameters and use these as initialization for the next iteration.

[9]The PTZ camera is held fixed in the experiments.

A. Intrinsic active triangulation calibration

The major focus of this work is calibrating a system that has many 3D groups. We can, however, apply our algorithm to a very basic level to calibrate the intrinsic parameters of a 3D group.

We would like to calibrate intrinsic parameters of an active triangulation sensor consisting a laser projector and a camera (camera 2 in Fig. 4). The intrinsic parameters of this system are described earlier in Section VIII-B.

We collected 10 images of a checkerboard (see Fig. 5) and 5 images of planes (see Fig. 6). We used 9 checkerboard images and 4 plane images for training and the rest for validation.

First, we considered incremental calibration which calibrates intrinsic, extrinsic incrementally. We used the Caltech calibration toolbox [1] to obtain the intrinsic parameters of the camera. We then held these intrinsic parameters fixed and calibrated other parameters of the active triangulation sensor. We call this 'Incremental calibration' based on the fact that we did not calibrate the intrinsic parameters of the camera jointly with other parameters.

Next, we considered our joint approach that optimizes all intrinsic parameters of the active triangulation system simultaneously and called the method 'Joint calibration'.

The results of different calibration methods on the validation set are reported in Table I. As can be seen from the table, the joint calibration technique with random restart gives very good result for calibration. A result of joint intrinsic calibration is illustrated in Fig. 7.

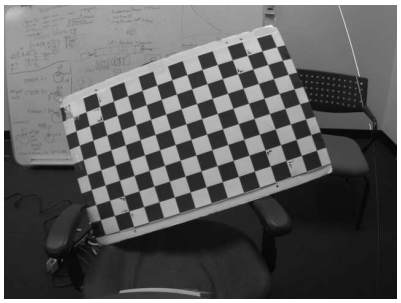

Fig. 5. An image of a (distorted) checkerboard and a laser beam captured by camera 2 (right corner).

Fig. 6. Three images of three planar regions captured by camera 2.

We have just showed that by a very simple and general method, we can calibrate the intrinsic parameters of an active triangulation system very accurately. We note that special techniques exist for calibrating these systems [34]. However, although their method [34] may produce better results, it is prohibitively more expensive in terms of human supervision and complex hardware compared to our method. In particular, they [34] use an LCD projector with coded line patterns for the sensor and granite reference plane with flatness of 7 microns for the calibration target. Nevertheless, we think smaller calibration error is not needed for robotic manipulation tasks because robot arms usually have repeatability of a millimeter.

TABLE I

MEAN VALIDATION ERROR OF ACTIVE TRIANGULATION CALIBRATION

Method	Error (mm)
Incremental calibration (without random restart)	4.11
Incremental calibration (with random restart)	1.83
Joint calibration (with random restart)	0.75

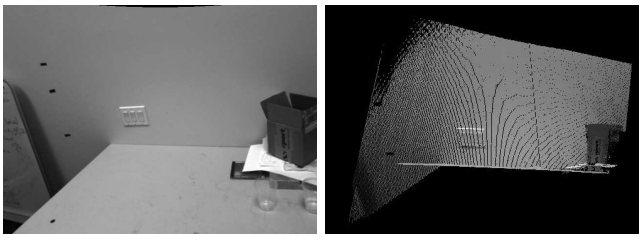

Fig. 7. Test result of intrinsic calibration of an active triangulation sensor. Left: Image taken by camera. Right: 3D data captured by the active sensor. The table and wall surfaces are flat thanks to very accurate calibration.

B. Simple calibration test

Over several years, we had made small changes to our Katana robot arm; as a result, the manufacturer supplied parameters were no longer accurate. Further, we found it difficult to directly measure these parameters (using a ruler, for example). To better understand the effects of these errors, we took the best calibration parameters of the active triangulation sensor in the previous section, and calibrated that against our arm, using the default link lengths and angle offsets provided by the manufacturer. We did this using the algorithm in [3], and global optimization using Levenberg-Marquardt. We obtained a calibration error of more than 3 cm on a validation set. For many manipulation tasks, this error is unacceptably high.

Since the active triangulation sensor has an error well under 3 cm (and the arm arm has high repeatability), this strongly suggests that the main source of error in this result is the inaccurate kinematic parameters. Here, kinematic parameters include link lengths, angle offsets, encoder offsets and angle offsets. It is very difficult to directly measure angle offsets or encoder offsets, so a better way to fix these parameters is by separately calibration the arm's kinematic parameters. We describe this in the next section.

C. Whole system calibration

In this experiment, we would like to calibrate the entire system at once. This includes the extrinsic parameters between sensors and intrinsic parameters of each sensor.

Throughout the experiment, we would like to assess the contribution different ideas mentioned earlier in the paper: joint intrinsic-extrinsic, sharing sensors, and redundancy. Candidates for comparison are incremental calibration, and joint calibration with some degrees of sharing and redundancy. These methods are described as follows.

First, we used an incremental approach to calibration. We calibrated the two cameras using the Caltech camera calibration toolbox [1], then calibrated an active triangulation system formed by grouping camera 1 and Laser. In the third step, we took the arm manufacturer's kinematic parameters, and then further adjusted them improve how well they match a set of of measured positions of the arm when placed in a variety of poses. Finally, we calibrated three systems together. This procedure is illustrated in Fig. 8.

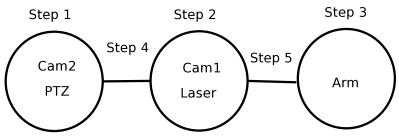

Fig. 8. Incremental calibration steps.

Next, we used our joint calibration approach and calibrated everything jointly using the method described in this paper. Since there are so many possible groupings. First, we decided to use the same groupings as before. The difference is that we will calibrate these two systems and the arm altogether in one optimization. This procedure is called 'Joint calibration 1' and illustrated in Fig. 9.

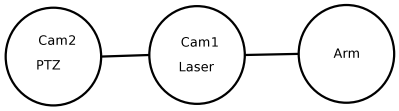

Fig. 9. Joint calibration 1: one-step calibration without sharing and without redundancy in extrinsic parameters.

Finally, as there are also other possible groupings, for example, we can group any pair of cameras or any camera and the laser projector. We decided to use all possible groupings between cameras and the projector. Such grouping has a property that some sensors are shared between many groups. The two procedures are called 'Joint calibration 2' (Fig. 10) and 'Joint calibration 3' (Fig. 11). The difference between these two methods is that 'Joint calibration 2' has a tree structure while 'Joint calibration 3' is a graph. Thus there is more redundancy in 'Joint calibration 3'.

We collected 10 checkerboard images with corresponding encoder readings for each corner of the checkerboard. We also collected 5 images of planes. To extensively evaluate calibration methods, we employed a five-fold cross validation scheme. Each fold contains 8 checkerboard images and 4 plane images for training and the rest for validating. Every calibration method was trained and validated five times.

Cross validation results of all the calibration methods are

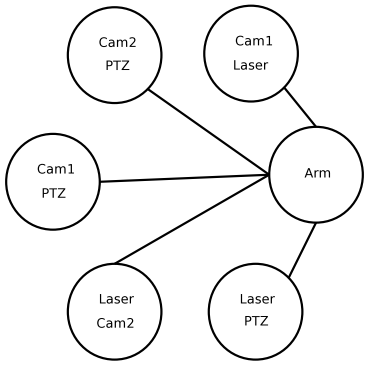

Fig. 10. Joint calibration 2: one-step calibration with sharing and some redundancy.

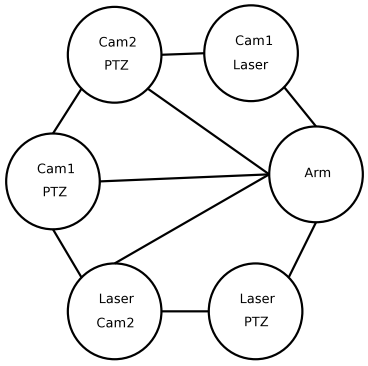

Fig. 11. Joint calibration 3: one-step calibration with sharing, more redundancy.

reported in Table II. [10] It can be seen from the table, our joint calibration approaches give significantly better results than the incremental approach. Also, joint calibration with more redundancy has higher accuracy. An explanation is that different pair sees different views of the world, the algorithm can make use of more constraints. Another reason is that joint calibration does not accumulate errors like in the case of calibrating in pairs.

TABLE II
MEAN/STD CROSS VALIDATION ERRORS OF WHOLE SYSTEM CALIBRATION.

Method	Error in calibration (mm)
Incremental calibration (one component at a time, Fig. 8)	24.76 ± 6.32
Joint calibration 1 (no sharing, Fig. 9)	5.19 ± 2.41
Joint calibration 2 (sharing, some redundancy, Fig. 10)	2.08 ± 1.27
Joint calibration 3 (sharing, more redundancy, Fig. 11)	1.17 ± 0.89

The incremental calibration method is more costly in human time because we have to work with one group at a time. In contrast, with our joint calibration technique, there is much less work on planning, code running. In total, it takes

[10]To compute the errors, we used geometric constraints on pair of sensor and arm and averaged errors all pairs.

us less than 2 hours of human time to get all the calibration parameters of the whole system.

Note that compared to the work of [16], [17], our method assumes parametric kinematic models of the arm. We chose this because the kinematic model is known, simple and accurate. For a comparison, the average errors presented in [17] are in the order of centimeters while our average errors are in the order of a millimeter.

XII. Conclusion

We considered joint calibration as a method to calibrate multiple sensors simultaneously. We built our framework upon 3D systems, geometric constraints and sharing sensors. Our experiment showed strong results for calibration of a robot that has many sensors.

Many robotic manipulation applications require calibration accuracy in the order of a millimeter. Examples of such tasks are opening doors, pressing elevator buttons, rotating thermometer buttons, picking up small objects [35], [36], [28]. The algorithm presented in this paper is our first successful method which fulfills the precision requirements for the tasks. Not only does it satisfy the accuracy requirement, this method also requires much less human time compared to older methods. The fact that it takes less time is significant because the arrangement of our sensors changes quickly over time and we would like to have a method that can calibrate everything as fast as possible.

Thanks to accurate calibration, we improved significantly the success rates of the above tasks in our STanford AI Robot (STAIR). Video segments of STAIR pressing elevator buttons, pulling drawers, rotating knobs using our method can be found in our project website (http://stair.stanford.edu). We note that in the videos, the hardware settings changed over time and thus having a fast and good approach for calibration is essential.

Acknowledgment

We thank Paul Baumstarck, Adam Coates, Timothy Hunter, Chetan Kalyan, Ellen Klingbeil, Morgan Quigley, Olga Russakovsky, Ashutosh Saxena, Ashley Wellman and STAIR teams for the help with the project and the paper. Support from the Office of Naval Research under MURI N000140710747 is gratefully acknowledged.

References

[1] J. Bouguet, "Caltech camera calibration toolbox for MATLAB," Caltech, Tech. Rep., 2008. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[2] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," CMU, Tech. Rep., 2005.

[3] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of Optical Society America*, vol. 4, 1986.

[4] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal on Robotics Automation*, vol. 3, no. 4, 1987.

[5] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *International Conference on Computer Vision*, 1999.

[6] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[7] H. Yamazoe, A. Utsumi, and S. Abe, "Multiple camera calibration with bundled optimization using silhouette geometry constraints," in *International Conference on Pattern Recognition*, 2006.

[8] Q. Zhang and R. Pless, "Extrinsic calibration of camera and laser range finder," in *IEEE/RSJ International Conference on intelligent robots and systems*, 2004.

[9] R. Bernhardt and S. Albright, *Robot calibration*. Kluwer, 1993.

[10] H. Zhuang, J. Yan, and O. Masory, "Calibration of stewart platforms and other parallel manipulators by minimizing inverse kinematic residuals," *Journal of Robotics Systems*, 1998.

[11] H. Zhuang, S. Motaghedi, and Z. Roth, "Robot calibration with planar constraints," in *International Conference on Robotics and Automation*, 1999.

[12] S. Besnard and W. Khalil, "Identifiable parameters for parallel robots kinematic calibration," in *International Conference on Robotics and Automation*, 2001.

[13] L. Beyer and J. Wulfsberg, "Practical robot calibration with ROSY," *Robotica*, vol. 22, pp. 505–512, 2004.

[14] H. Zhuang and Z. Roth, *Camera-aided robot calibration*. CRC-Press, 1996.

[15] N. Roy and S. Thrun, "Online self-calibration for mobile robots," in *International Conference Robotics and Automation*, 1999.

[16] J. Kolter and A. Ng, "Learning omnidirectional path following using dimensionality reduction," in *Robotics Science and Systems*, 2007.

[17] J. Sturm, C. Plagemann, and W. Burgard, "Adaptive body scheme models for robust robotic manipulation," in *Robotics Science and Systems*, 2008.

[18] R. Y. Tsai and R. Lenz, "Real time versatile robotics hand/eye calibration using 3D machine vision," in *International Conference on Robotics and Automation*, 1988.

[19] R. Y. Tsai, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, 1989.

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

[21] K. Daniilidis, "Hand-eye calibration using dual quaternions," *International Journal of Robotics Research*, 1999.

[22] H. Zhuang and K. Wang, "Simultaneous calibration of a robot and a hand-mounted camera," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 5, 1995.

[23] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Transactions on Robotics and Automation*, 1998.

[24] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Johns Hopkins Press, 1996.

[25] I. Inc, "The opencv opensource computer vision library," Intel Inc., Tech. Rep., 2009. [Online]. Available: http://opencvlibrary.sourceforge.net

[26] D. Forsyth and J. Ponce, *Computer Vision - A Modern Approach*. Prentice Hall, 2003.

[27] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[28] M. Quigley, S. Batra, S. Gould, E. Klingbeil, Q. Le, A. Wellman, and A. Y. Ng, "High accuracy 3D sensing for mobile manipulators: Improving object detection and door opening," in *International Conference on Robotics and Automation*, 2009.

[29] N. Pears and P. Probert, "Active triangulation rangefinder design for mobile robots," in *IEEE/RSJ International Conference on intelligent robots and systems*, 1992.

[30] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons, Inc, 2006.

[31] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *The Quarterly of Applied Math*, vol. 164, 1944.

[32] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 431, 1963.

[33] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2006.

[34] J. Guehring, C. Brenner, J. Bhm, and D. Fritsch, "Data processing and calibration of a cross-pattern stripe projector," in *ISPRS Congress*, 2000.

[35] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *International Conference on Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[36] E. Klingbeil, A. Saxena, and A. Y. Ng, "Learning to open new doors," in *Robotics Science and Systems (RSS) workshop on Robot Manipulation*, 2008.