Extrinisic Calibration of a Camera-Arm System Through Rotation Identification

Steve McGuire*, Christoffer Heckman, Daniel Szafir, Simon Julier, and Nisar Ahmed*

Abstract—Determining extrinsic calibration parameters is a necessity in any robotic system composed of actuators and cameras. Once a system is outside the lab environment, parameters must be determined without relying on outside artifacts such as calibration targets. We propose a method that relies on structured motion of an observed arm to recover extrinsic calibration parameters. Our method combines known arm kinematics with observations of conics in the image plane to calculate maximum-likelihood estimates for calibration extrinsics. This method is validated in simulation and tested against a real-world model, yielding results consistent with ruler-based estimates. Our method shows promise for estimating the pose of a camera relative to an articulated arm's end effector without requiring tedious measurements or external artifacts.

Index Terms—robotics, hand-eye problem, self-calibration, structure from motion

I. INTRODUCTION

In robotics, data fusion between multiple sensors is frequently required in order to accomplish some task in an operating environment. Often, the relative locations between various components must be known to a high degree of precision; any error in relative pose propagates throughout the system without possibility for correction. These locations can be determined via idealized means, such as computer-aided design, or estimated means, such as motion capture systems or tag trackers. For many sensor types, the precise point of measurement may be buried within a housing or even within an integrated circuit, making direct measurement impractical. Furthermore, such idealized measurements may fail to capture assembly variations or installation error.

In this work we consider the goal of calibrating an assembled group of kinematically linked sensors and actuators as shown in Fig. 1. To support the idea of "bolt together and go" robotics, calibration should be determined by operating the assembly in its environment and analyzing the associated output without using calibration targets or precision artifacts. Such scenarios arise frequently in the operation of robotic arms, boom cameras, and other articulated sensor mounts used for capturing scientific data. A significant barrier in such operations is measuring the exact relative transforms between various sections of the kinematic chain, such as the transform from an imaging sensor to an articulated end effector. We develop an approach to calibration that does not

S. McGuire and N. Ahmed are with the Department of Aerospace Engineering Sciences, University of Colorado at Boulder. C. Heckman, D. Szafir, and G. Sibley are with the Department of Computer Science, University of Colorado at Boulder. S. Julier is with the Department of Computer Science, University College London.

* Corresponding authors.

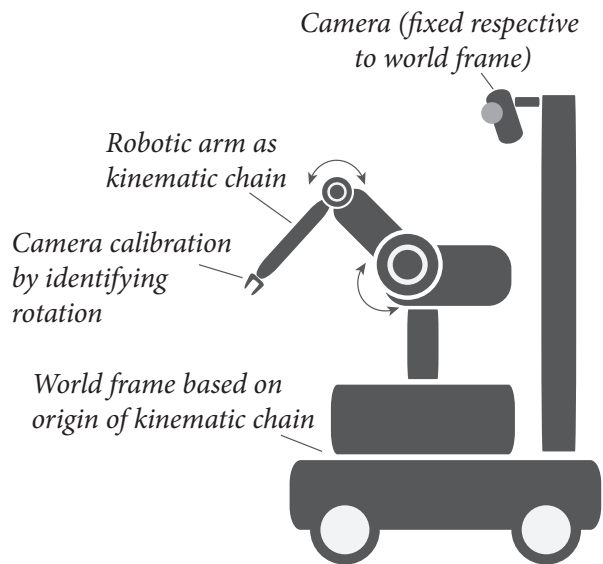E-mail: [stephen.mcguire,nisar.ahmed]@colorado.edu

Fig. 1: Overview of the hand-eye calibration problem on a mobile robot base. A fixed camera rigidly mounted to the base of a robot arm observes the arm's end effector movements.

rely on highly precise measurements for difficult-to-determine quantities; instead, we use a camera to observe the arm's motion and estimate the required quantities.

Our approach estimates relative poses between an actuator and camera by observing features on the actuator under structured motion. These features trace out circles in 3-D world space; we estimate a 3-D model to be projected into the image plane in order to find relative poses between an actuator and an observing camera. The present work is the first application of the tracking of features to model 3-D circles in space, rather than modeling elliptical paths in projective space. This approach has many advantages related to noise robustness as described in Section III, and combines significant achievements in structure from motion, shape estimation, camera pose estimation, and self-calibration.

The specific problem considered here is to determine extrinsic calibration parameters for a camera-arm system, such that the camera has a view of the end-effector of the arm, as shown in Fig. 2, with an idealized sketch in Fig. 3. An arm has i revolute joints yielding one degree of freedom per joint. Each arm joint produces a measurement θ_i, representing the angular deflection of the revolute joint i. The arm has a well-known kinematic model, assumed to be precision manufactured. For this work, the final joint in the arm is a wrist-type joint with a rigidly-mounted end effector. The camera is assumed to be calibrated for intrinsic parameters

Fig. 2: Real-world setup of the calibration problem showing a fixed camera rigidly mounted with respect to a robotic arm. In this experiment, a six-jointed Kinova Jaco2 arm is used.
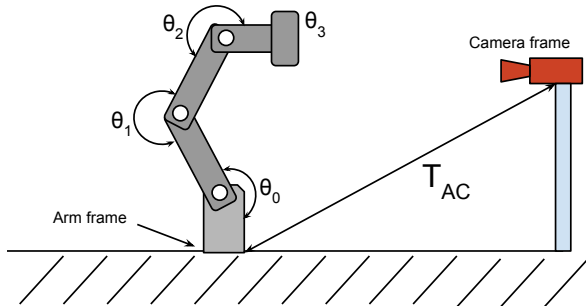

Fig. 3: Simplified sketch of setup. In this setup, the world frame is defined as the origin of the arm's kinematic chain. The camera is fixed with respect to the arm's kinematic origin via rigid transform T_{AC}. Note that our method does not assume coplanar rotations as suggested by the diagram.

via other means (e.g. [1]) and rigidly mounted via a fixed, but unknown, transform T_{AC} to the origin of the arm's kinematic chain.

II. RELATED WORK

Camera-arm calibration is typically referred to as the "hand-eye' problem in robotics and may be specified with either a stationary camera observing the motion of an arm or a mobile camera rigidly mounted to the end effector of an arm. An example of the current state of the art is from Pradeep et al. [2], where precision calibration targets are moved via well-known robot kinematic models to recover relative transforms between cameras and the known kinematic models. Our work develops a technique for recovering these same transforms without the need for precision calibration targets by exploiting how a camera image changes under robot motion. Specifically, we are interested in wrist-like rotary motions. Through this type of arm motion, we can factor the hand-eye problem into several component problems: feature detection, feature tracking, geometry reconstruction, and finally pose reconstruction. Our work focuses on the geometry reconstruction and the pose reconstruction aspects of the factored hand-eye problem.

In our formulation of rotational geometry reconstruction, we are interested in recovering the parameters of a 3D circle from a set of observed image points. This is possible by leveraging an observation from Jiang et al. [3], who originally established a method for reconstruction of a moving object with respect to a fixed camera under the assumption that the motion is constrained to rotation about a single axis. In that work, it was observed that individual points rotating about a single axis will trace out an ellipse under projection. The main focus of [3] relates to reconstruction from a single rotation; we use multiple independent rotations to establish the relative transform from the target object to the camera.

In [4], Sawhney et al. describe a method for reconstruction of objects under rotational motion, assuming that the camera undergoes the rotational motion attached to an arm observing a static scene. Sawhney develops much of the mathematical framework that will be used in our method. In [5], Fremont and Chellali address how to create a 3-D reconstruction of an object by rotating it about an axis, but does not address how to estimate the pose of such an object with respect to an arbitrary image plane. In [6], Liu and Hu use a fixed camera to observe a cylindrical spacecraft and estimate pose using a known CAD model to match ellipses to metric features. Their work relies on static imagery and resolves scale using the CAD model; in our formulation, we have no information about the configuration of the end effector.

Hutter and Brewer [7] used the approach of approximating elliptical shapes in the image plane corresponding to true circles in 3-space [4], [8], [9] to estimate the pose of vehicle wheels from segmented images for self-driving car applications. Their work recovers the wheel's rotation axis in order to estimate steering angle. Our work accomplishes a parallel goal of recovering the end effector's rotation axis in order to estimate properties of the larger system. Another application is in [10], where pose estimation from an image of a well-known ellipse is used to calibrate a laser-camera system. However, the parameters of the underlying circle must be known a priori in the form of a known calibration target.

Lundberg et al. [11] pose a similar problem setup, but use a single well-known feature on the end effector. This well-known feature is precisely positioned in each of a series of frames to form a 'virtual' calibration target. A calibration target is thus assembled in a point-wise manner, allowing calibration to continue using conventional techniques. A key difference between their solution and ours is the uniqueness and the prior knowledge of the well-known feature; our solution can utilize ambient features discovered at runtime.

Also connected is that of Forsyth et al. [12], where camera pose was estimated by observing images of well-known conics. In contrast, we observe conics by tracking features of a rotating object over time; a critical distinction is that the world geometry of the observed conics is not well-known. Borghese et al. [13] also applied Forsyth's work for pose estimation, but only applied this to rotated calibration targets and not to generated conic paths in 3-D space resulting from object rotation.

III. PROCEDURE

The goal of our method is to estimate the camera's location relative to the base of the arm's kinematic chain. First, images are captured of the end effector rotating about the axis of the last joint in the arm while the remainder of the arm is motionless. While the arm is rotating, feature points (described in Sec. III-A) on the end effector are tracked and associated between frames to produce a set of arcs in the image plane. During an initial optimization step, the observed axis of rotation is estimated by requiring that each arc's center of rotation be collinear, with independent radii and distance along the rotation axis from an arbitrary starting point. Meanwhile, the arm's joint angles θ_i are captured to calculate the arm's expected axis of rotation in the world frame. This procedure is repeated a number of times n, where the arm is repositioned to a different pose and the process is repeated. After all measurements are collected, the position of the camera in the arm frame is estimated. This technique is presented in video form in the Supplementary Materials.

We use the encoding of a pose $\boldsymbol{p} \in \mathbb{R}^6$ as composed of six elements: $\left(x, y, z, \phi, \theta, \psi\right)^T$ (the latter three quantities representing roll, pitch, and yaw respectively). Each measurement observes the x, y position in the image plane of m feature points, as well as the arm angular measurements θ_i. The set of all unique observed feature points is L. A 4×4 homogeneous transform giving the location of object A in coordinate frame B is denoted $\boldsymbol{T}_{BA}$. Estimates of a particular quantity P are denoted by $\hat{P}$. Parameters to be estimated comprise the pose $\boldsymbol{p}$ corresponding to the location of the camera in the arm frame, represented alternatively as $\boldsymbol{p}_{AC} \in \mathbb{R}^6$ or $\boldsymbol{T}_{AC}$ as a 4×4 homogeneous transformation. These two representations will be used interchangeably via the Lie SE(3) $\exp()$ and $\log()$ operators where needed.

The estimation procedure follows several steps: 1) arc identification, 2) circle estimation, 3) estimation of the rotation axis by vision, 4) rotation axis estimation using forward kinematics, and 5) estimation of $\hat{T}_{AC}$.

A. Arc Identification

To establish arcs, ambient features on the end effector are tracked across frames. The choice of feature descriptor is arbitrary, so long as the end effector can be tracked across more than three frames to be able to fit a projection of a 3-D circle to the track. The reference implementation utilizes OpenCV's simple blob detector as input to the Lucas-Kanade optical flow algorithm to obviate the effect of local minima. Tracks that are shorter than a minimum threshold are discarded, as are tracks that do not encode sufficient motion between frames. As features potentially rotate out of view, new features are detected as potential tracks. The use of ambient features forms the novelty of our work, eliminating the need for precision targets and/or well-known features.

B. Rotation axis identification (vision)

Once tracks are assembled, an initial optimization step is performed to recover the shared rotation axis of the generated ellipses. The choice of coordinate axes corresponds to the image frame, with x right, y down, and z forward. In this optimization step, the parameter vector $\mathbf{l}_k$ consists of $5+2m$ terms: an (x, y, z) point on a 3-D line, angles θ and ϕ indicating the rotation along XY and YZ axes, and a radius from the line and displacement from the origin point for each ellipse to be fit. This parameterization enforces a coaxial constraint such that the centers of every circle in 3-D are collinear. In this stage, the camera is set at $(0, 0, -1, 0, 0, 0)$ with respect to the origin. Initial conditions for the 3-D circle optimization are set to $(0, 0, 0, 0, 0)$, with each potential arc having incremental displacement and unit radius. The coordinates to be recovered are in an arbitrarily transformed up to a scale space.

Candidate points on circles corresponding to the parameter vector are then generated and projected into the image plane according to the given camera model's projection matrix. For each detected feature's arc, an ellipse is then fit to these projected points belonging to the parameter vector's 3D circle using a direct least squares method [14]. A residual is then composed by summing the distance between projected points and the candidate ellipse in projective space for each feature track. The use of a coaxial constraint on the underlying model used to generate fit ellipses aids in noise robustness, as a poor track can corrupt an individual ellipse fit.

For each measurement k, the candidate parameters $\mathbf{l}_k$ identify a line in 3-D; points are generated along this line, projected into the image plane, and a best-fit line in 2-D is recovered; this line represents the projection of the axis of rotation into the image plane.

These lines are stored in point-slope form, yielding an observation $z_k = (m_k, b_k)$.

C. Rotation axis identification (arm)

Given that the kinematic chain of the arm is provided, without loss of generality we consider a single joint of the arm. With a revolute joint, the transition through the joint is characterized by seven parameters: six to identify the fixed mechanism of the arm leading into the joint (denoted link L), and one to adjust the output face based on actuator angle (denoted T_{θ_i}). Each transform is represented in the parent joint's coordinate system; for joint i, the kinematic chain is therefore represented in the coordinate system of joint $i - 1$ as:

$$T_{(i-1)(i)} = L_i T_{\theta_i} \tag{1}$$

Note that T_{θ_i} only encodes a single rotation; the axis of rotation of joint i. Therefore, any points of the form $\left(0 \; 0 \; z'\right)^T$ will lie on the rotation axis of joint i, in joint i's coordinate frame. World coordinates of a set of such points may be obtained by applying the arm's kinematic chain using homogeneous transformations.

D. Camera to arm rigid transform estimation

The estimated rigid transformation between the camera and the base of the arm $\hat{\boldsymbol{T}}_{AC}$ is estimated by applying bundle adjustment over all measurements n.

Modeling the image noise as zero-mean Gaussian results in a maximum-likelihood estimator, which can be stated as a nonlinear least squares optimization problem of the form

$$r = e(z, x)R^{-1}e(z, x))^T \qquad (2)$$

where $e(z, x)$ is function of observations z and model parameters x that produces a residual error vector, and R is a block-matrix of weights in the projected axis from the vision system corresponding to the uncertainties in the axis recovery. For this problem, the parameters x to be recovered form a vector $(x, y, z, \phi, \theta, \psi)^T$ representing the $\mathbb{R}^6$ form of $\hat{T}_{AC}$, while the observations z are the stacked coefficients of the line fits $z_k = (m_k, b_k), k = 1...n$ describing the axis of rotation for each of n measurements. Since the recovered projected centers corresponding to each feature track are each located some unknown distance from the origin of the joint immediately prior to the end effector, we cannot directly compare known world geometry of the end effector to the observed axes of rotation. However, we exploit the fact that the true axis of rotation must be collinear with the observed axis of rotation to form a residual function e.

For each of n measurements k, several test points of the form $(0\ 0\ z')^T$ are generated in the last joint's coordinate frame. A minimum of two test points are required to define the line representing the projection of the arm's axis of rotation in the image plane. While more points can be used, the assumption of a rectified camera model guarantees that any additional points will be collinear. These j test points are first transformed into world coordinates using the arm's kinematic model and then projected into the image plane using the current parameter vector x. For each measurement k, the previously identified line parameters $(m_k\ b_k)$, the distance $d_{k,j}$ between each projected test point $t_j = (u_j\ v_j)$ and its associated projected axis of rotation a_k is calculated by:

$$d_{k,j} = \frac{-m_k u_j + v_j - b_k}{\sqrt{m_k^2 + 1}} \qquad (3)$$

as given in [15].

The distance metric of Equation 3 is used to assemble the residual vector r by concatenating over all test points and all measurements:

$$r = (d_{11}\ d_{12}\ ...\ d_{1j}\ d_{21}\ d_{22}\ ...\ d_{2j}\ ...\ d_{n1}\ d_{n2}\ ...\ d_{nj})^T \qquad (4)$$

We implement our optimization using Levenberg-Marquardt with random restarts to avoid local minima.

To avoid numerical instability resulting from the use of finite differences, we apply the technique of [16] to determine the value of the Jacobian at the estimate $\hat{x}$, defined as

$$\mathbf{J} = \left.\frac{\partial \mathbf{r}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}} \qquad (5)$$

Once $\mathbf{J}$ is determined, the diagonal terms of $(\boldsymbol{J}^T\boldsymbol{J})^{-1}$ estimate the variance in each recovered component of $\hat{\boldsymbol{T}}_{AC}$.
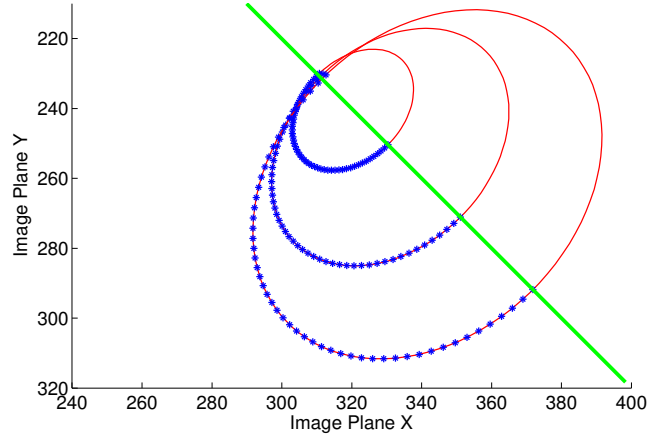

Fig. 4: Simulated projected view of a set of three feature tracks. Source track data are in blue stars, fitted ellipses in solid red, and projected axis of rotation through circle centers in green.

E. Measures

We evaluate our simulation results through two sets of measures. The first set evaluates the quality of the rotation axis identification from feature track points, while the second evaluates the quality of the camera pose reconstruction. Rotation axis identification quality is measured by relative error in the 2D line parameters $(m_k, b_k)^T$ for each measurement k; these errors are expressed in pixels^{-1} and pixels respectively. The camera pose reconstruction quality is measured by the relative error in 3D position parameters $(x, y, z, \phi, \theta, \psi)^T$, expressed in units of meters and radians.

IV. RESULTS

A. Simulation

In simulation, observed arc data are generated against a given camera location and arm end effector positions. An example of a simulated camera view is Figure 4, showing the progression from source data to the final measurement that will be used in the optimization, a line through the projected circle centers.

To characterize how tracking error propagates through the estimation technique, a Monte Carlo simulation was run at four different levels. Since two stages of optimization are present, we present results at both the ellipse fit stage and the final camera position fit stage. Zero-mean Gaussian noise at $\sigma^2 = \{0.1, 0.5, 1.0, 1.5\}$ px^2 was added to the projection of simulated ellipse points into the camera image plane. Six rotation observations were simulated; the error distributions are shown in Fig. 5.

Each of these simulation runs was then pushed through the second optimization step to recover the 3D pose of the camera. The error distributions in the Cartesian directions are shown in Fig. 6 and Fig. 7, while the angular errors are shown in Fig. 8 and Fig. 9.
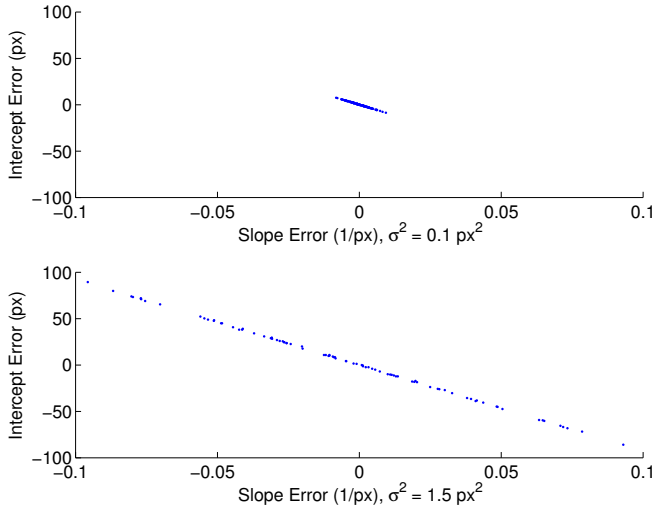
Fig. 5: Error distribution of rotation axis fits in with $\sigma^2 = 0.1$ px^2 and $\sigma^2 = 1.5$ px^2 noise showing error correlation and impact of increased tracking noise
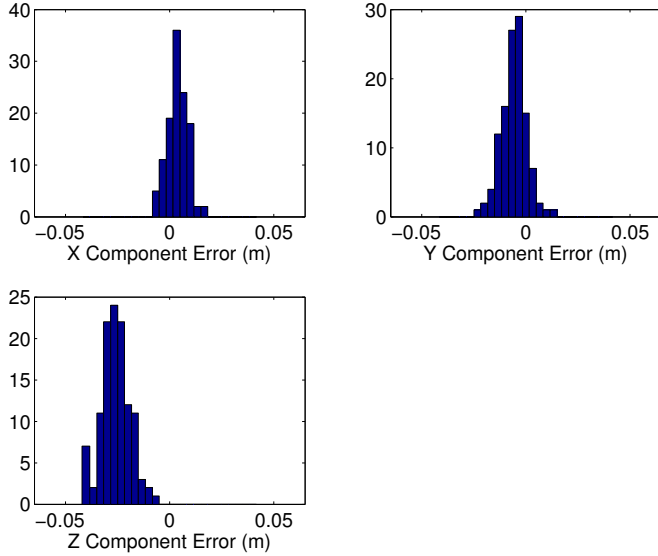

Fig. 7: Cartesian position error distribution with $\sigma^2 = 1.5$ px^2 noise

Fig. 6: Cartesian position error distribution with $\sigma^2 = 0.1$ px^2 noise

Fig. 8: Angular error distribution with $\sigma^2 = 0.1$ px^2 noise

B. Physical System

The system was tested using an Asus Xtion Pro RGB-D camera and a Kinova Jaco2 arm, shown in Figure 2. Calibration of camera intrinsics occurred offline, while the Kinova Jaco2 forward kinematics were derived from the manufacturer-provided model. The arm's end effector was equipped with a feature-rich covering to ensure adequate ambient features are available to be tracked, shown in Fig. 10. As features were tracked in the image, track data was assembled. In the real system, track data is not as smooth as the simulation; additional processing was implemented to detect jumps in the track that were inconsistent with a smooth arc. A minimum track length was established to remove poor tracks. Measurements were collected independently, with the estimation running offline. Each measurement consisted of approximately sixty seconds of rotation in the wrist joint,
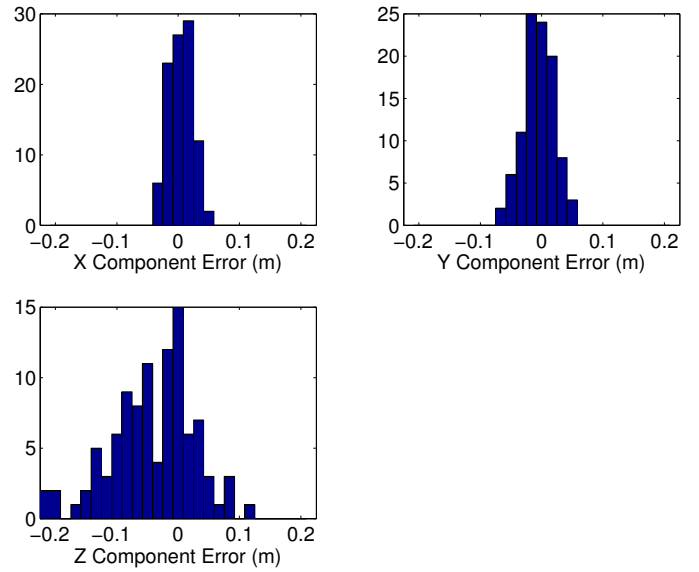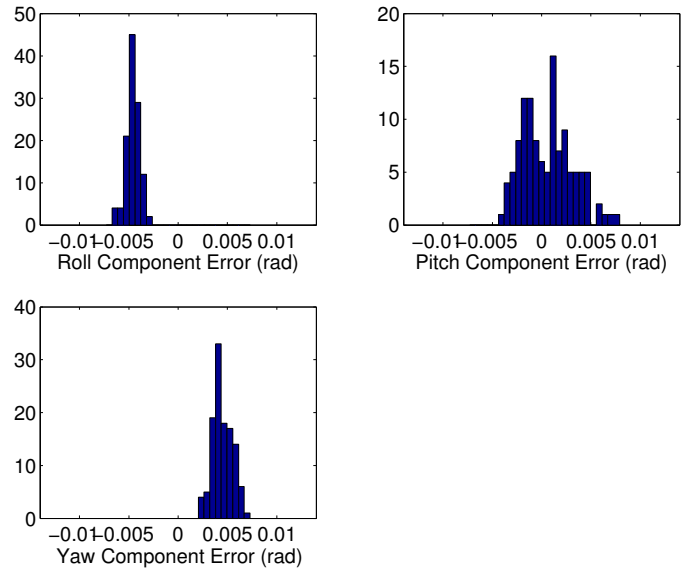
followed by an arm reposition for the next measurement.

After optimization completed, we were able to recover a pose consistent with ruler-based estimates of $\hat{T}_{AC}$. Our covariance estimate reported errors on the order of 0.1mm; these results appear to be excessively optimistic, given the relatively low quality of arc tracks used as observations. Reasons for this optimism could include the omission of error weights in the estimation of $(J^T J)^{-1}$, implying that all measurements are equally trustworthy.

V. DISCUSSION

In simulation, our technique was able to recover reasonably accurate estimates of camera position even in the presence
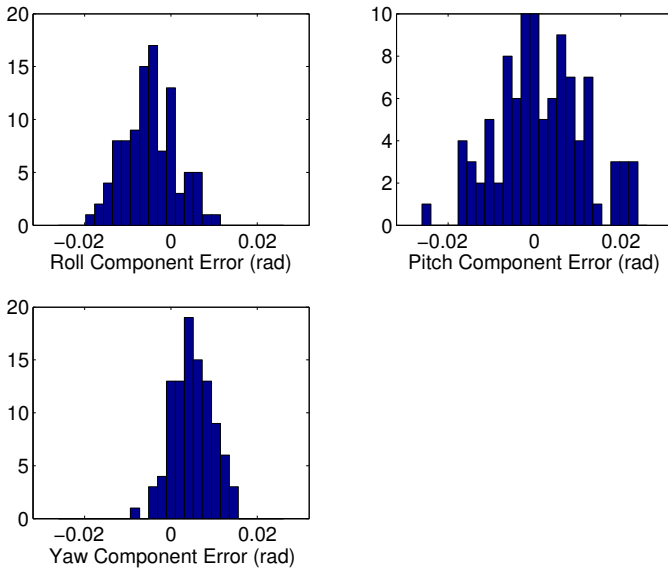
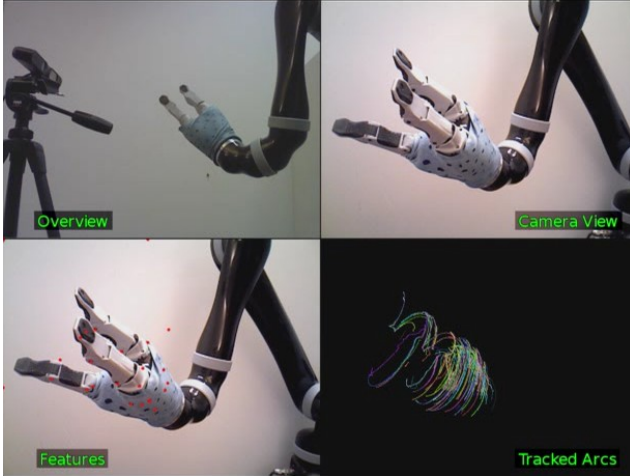Fig. 9: Angular error distribution with $\sigma^2 = 1.5$ px^2 noise

Fig. 10: Four views of the real system under test, showing an overview, the camera view, the detected features, and the tracked arcs

of tracking noise. Given the minimal time required to gather a calibration dataset and modest algorithm execution time (approx 5 minutes with a Matlab implementation), our technique seems practical to implement in real-world scenarios. We were able to demonstrate the function of our system aboard a representative real system with varying success.

In the line fit distributions, particularly at higher noise levels such as Fig. 5, we observe that the error terms are strongly correlated; we hypothesize that this correlation is due to the external constraint that all ellipses be coaxial. The noise characteristics appear to be dependent on the viewing angle of the camera with respect to the rotation axis of the arm. When recovering the line fit parameters for the measurement data associated with the wider distributions in Fig. 5, the optimizer terminated with a summed residual value several orders of magnitude higher than the summed residuals of the

narrow distributions. By treating large residuals as outliers, the overall robustness of the estimation technique could be improved.

When the line fit errors are propagated through to the camera pose recovery step, we note that several position estimates appear to be biased, for example, the Z component error in Fig. 6. These error propagation results lead us to conclude that an uncorrelated and unbiased Gaussian error model in output noise is not appropriate for this estimation technique. The variance of the expected errors are within several centimeters even at the higher noise level. Based on these simulation results, we believe that this technique is worth further investigation.

While our method has shown promising results, several important areas remain for further investigation. The effect of each of these areas on our results is demonstrated in the Supplementary Materials.

A. Measurement constraints

Several conditions on arm pose have been identified as necessary (but not sufficient) to have a convergent optimization. For example, in a single crossing scenario, the end effector of the arm was commanded to have the same $\begin{pmatrix} x & y & z \end{pmatrix}$ position with different orientation values. This setup represents a degenerate pathological problem because the location of the camera is underdetermined. Possible locations for the camera lay on a line in 3D space perpendicular to the 2D projection of the crossing point; since the residual computation only considers distance from a test point to the associated rotation axis, all camera positions yield the same residual value. Camera orientation can be recovered, but position cannot. To avoid this condition, a minimum of three measurements are required, with one measurement having a different position from the other two.

B. Differences between simulation and real robot

The performance under simulation appears to recover the camera position more accurately than the use of a real system; we have several hypotheses to explore these differences. In the real system, a physical camera was used with a basic linear camera model that fails to completely replicate the ideal camera used in simulation. There is also quantization error in the camera vs simulation, particularly in the generation of arc portions used in the ellipse fit stage. In simulation, these arc points are carried through with double-precision floating point values throughout; while subpixel resolution techniques in real imagery can help to mitigate this effect, further work is required to improve the fidelity of the simulation. We studied error in tracking by introducing zero-mean independent Gaussian noise into the arc track data image coordinates; observing the real tracks produced in our reference implementation, shown in Fig. 10, a Gaussian noise model may not be appropriate.

C. Optimization initial values

In both stages of optimization, testing has revealed a significant dependency on initial conditions to the optimizer

to yield a convergent solution due to degeneracies in the 3D circle fitting procedure; this phenomenon is very common in SLAM algorithms as noted by [17]. Since the residual is defined as the distance between the projected ideal circle and the observed arc track, there is no analysis of curvature to ensure that the projected ideal circle is approximately oriented at initial evaluation of the residual function.

D. Known kinematics

This approach relies on the manufacturer-provided model of the arm kinematics in order to resolve the rotation axis at the end effector with respect to the base of the arm as truth. This assumption could be improved by adding optimization parameters such as angular encoder bias to the various joints of the arm, to allow for installation errors. Other types of deviations from the manufacturer model, such as damage or wear, offer significant challenges in modeling but might be amenable to an online error analysis such as in [1].

E. Precalibrated camera

For our optimization to be successful, we require known camera parameters and linearly rectified camera data as input. An important observation is that monocular self-calibration techniques such as [18] and our technique are not exclusive; as self-calibration relies on motion in the environment and the arm is rigidly mounted to the camera, the arm will appear as a static obstruction. This requires the arm to be held stationary during calibration, while the reference implementation of the arc tracker requires the background to be stationary.

F. Tracker noise

In the reference implementation, a basic tracker was used that did not attempt to enforce any motion dynamics on the detected tracks. An improvement on this work would include integrating a more advanced tracker that can incorporate a motion model to smooth out detected arcs and make the tracker more robust to noisy image data.

G. Path planning for calibration poses

An important unanswered question in our technique is the method of determining what constitutes a 'good' pose for use in calibration. Ideally, a statistical metric could be determined to evaluate the information content of a new pose given the set of existing poses; this metric could then be used to develop a path planner that identifies a series of most valuable poses for reducing error in the final estimate of $\hat{\boldsymbol{T}}_{AC}$ under constraints of robot kinematics and camera field of view.

H. Coordinate system consistency

While we have the convenience of defining our coordinate frames without regard to external needs, a real system must take into account a wide range of sensor data that may or may not even use the same coordinate or rotation convention. Multiple coordinate system standards pose a hazard to interpretation and development of calibration routines and reduce the generality of our technique.

I. Incorporating measurement weights

In future work, we envision using the residual value of the rotation axis identification stage as a means of weighting the relative value of that measurement in the pose reconstruction stage. The residuals of all measurements could be set on a scale to aid in outlier rejection, such that poor tracking in one particular measurement does not contaminate the overall result.

VI. Conclusions

We have presented a method for calibrating an articulated arm with a wrist joint to a camera without requiring calibration targets, instead relying on structured arm motion and ambient features on the end effector. Our method requires no knowledge of the mechanism of the end effector or unique features to be present. We have validated our results in simulation and demonstrated with real-world data yielding promising results. A Monte Carlo analysis of error propagation verifies that small errors in the quality of feature tracks do not cause the resulting position estimates to degrade dramatically. Further testing is required to explore system degeneracies and validate performance in realistic environments.

VII. Acknowledgments

References

[1] N. Keivan and G. Sibley, "Asynchronous adaptive conditioning for visual-inertial SLAM," *I. J. Robotic Res.*, vol. 34, no. 13, pp. 1573–1589, 2015.

[2] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach," in *Experimental Robotics*. Springer, 2014, pp. 211–225.

[3] G. Jiang, H.-t. Tsui, L. Quan, and A. Zisserman, "Geometry of single axis motions using conic fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1343–1348, 2003.

[4] H. S. Sawhney, J. Oliensis, and A. R. Hanson, "Description and reconstruction from image trajectories of rotational motion," in *Computer Vision, 1990. Proceedings, Third International Conference on*. IEEE, 1990, pp. 494–498.

[5] F. Marinello, P. Bariani, E. Savio, A. Horsewell, and L. De Chiffre, "Critical factors in sem 3d stereo microscopy," *Measurement Science and Technology*, vol. 19, no. 6, p. 065705, 2008.

[6] C. Liu and W. Hu, "Relative pose estimation for cylinder-shaped spacecrafts using single image," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 50, no. 4, pp. 3036–3056, 2014.

[7] M. Hutter and N. Brewer, "Matching 2-d ellipses to 3-d circles with application to vehicle pose estimation," *arXiv preprint arXiv:0912.3589*, 2009.

[8] K. Kanatani and W. Liu, "3d interpretation of conics and orthogonality," *CVGIP: Image understanding*, vol. 58, no. 3, pp. 286–301, 1993.

[9] Q. Chen, H. Wu, and T. Wada, "Camera calibration with two arbitrary coplanar circles," in *Computer Vision-ECCV 2004*. Springer, 2004, pp. 521–532.

[10] H. Alismail, L. D. Baker, and B. Browning, "Automatic calibration of a range sensor and camera system," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*. IEEE, 2012, pp. 286–292.

[11] I. Lundberg, M. Bjorkman, and P. Ogren, "Intrinsic camera and hand-eye calibration for a robot vision system using a point marker," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 59–66.

[12] D. Forsyth, J. L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell, "Invariant descriptors for 3d object recognition and pose," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 10, pp. 971–991, 1991.

[13] I. Frosio, A. Alzati, M. Bertolini, C. Turrini, and N. A. Borghese, "Linear pose estimate from corresponding conics," *Pattern Recognition*, vol. 45, no. 12, pp. 4169–4181, 2012.

[14] M. Fitzgibbon, A. W.and Pilu and R. B. Fisher, "Direct least-squares fitting of ellipses," vol. 21, no. 5, pp. 476–480, May 1999.

[15] E. W. Weisstein, "Point-line distance–2-dimensional," http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html, accessed: 2016-03-06.

[16] J. R. Martins, P. Sturdza, and J. J. Alonso, "The complex-step derivative approximation," *ACM Transactions on Mathematical Software (TOMS)*, vol. 29, no. 3, pp. 245–262, 2003.

[17] P. Newman and K. Ho, "Slam-loop closing with visually salient features," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 635–642.

[18] N. Keivan and G. Sibley, "Online SLAM with any-time self-calibration and automatic change detection," in *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*, 2015, pp. 5775–5782.