

Extrinsic Calibration from Per-Sensor Egomotion

Jonathan Brookshire and Seth Teller

MIT Computer Science and Artificial Intelligence Laboratory, {jbrooksh, teller}@csail.mit.edu

Abstract—We show how to recover the 6-DOF transform between two sensors mounted rigidly on a moving body, a form of extrinsic calibration useful for data fusion. Our algorithm takes noisy, per-sensor incremental egomotion observations (i.e., incremental poses) as input and produces as output an estimate of the maximum-likelihood 6-DOF calibration relating the sensors and accompanying uncertainty.

The 6-DOF transformation sought can be represented effectively as a unit dual quaternion with 8 parameters subject to two constraints. Noise is explicitly modeled (via the Lie algebra), yielding a constrained Fisher Information Matrix and Cramer-Rao Lower Bound. The result is an analysis of motion degeneracy and a singularity-free optimization procedure.

The method requires only that the sensors travel together along a motion path that is non-degenerate. It does not require that the sensors be synchronized, have overlapping fields of view, or observe common features. It does not require construction of a global reference frame or solving SLAM. In practice, from hand-held motion of RGB-D cameras, the method recovered inter-camera calibrations accurate to within ~ 0.014 m and ~ 0.022 radians (about 1 cm and 1 degree).

I. INTRODUCTION

Extrinsic calibration – the 6-DOF rigid body transform relating sensor coordinate frames – is useful for data fusion. For example, point clouds arising from different range sensors can be aligned by transforming one sensor’s data into another sensor’s frame, or all sensor data into a common body frame.

We show that inter-sensor calibration and an uncertainty estimate can be accurately and efficiently recovered from incremental poses (and uncertainties) observed by each sensor. Fig. 1 shows sensors r and s , each able to observe its own incremental motion v_{ri} and v_{si} respectively, such that the calibration k satisfies:

$$v_{si} = g(v_{ri}, k) \quad (1)$$

where g simply transforms one set of motions according to k .

This paper describes an algorithm to find the k that best aligns two series of observed incremental motions. The algorithm takes as input two sets of 6-DOF incremental pose observations and a 6×6 covariance matrix associated with each incremental pose. It produces as output an estimate of the 6-DOF calibration, and a Cramer-Rao lower bound (CRLB) [1] on the uncertainty of that estimate. (For source code and data, see <http://rvsn.csail.mit.edu/calibration3d>.)

Prior to describing our estimation method, we confirm that the calibration is in general *observable*, i.e. that there is sufficient information in the observations to define k uniquely. Observability analysis yields a characterization of singularities in the Fisher Information Matrix (FIM) arising from non-generic sensor motion. For example, singularity analysis reveals that 6-DOF calibration can not be recovered from planar-only motion, or when the sensors rotate only around a single

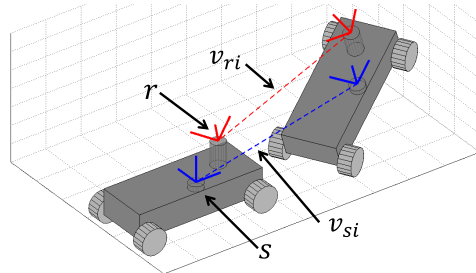


Fig. 1: The incremental motions of the r , red, and s , blue, sensors are used to recover the calibration between the sensors as the robot moves. The dotted lines suggest the incremental motions, v_{ri} and v_{si} , for sensors r and s , respectively.

axis. This confirms previous findings [3, 20] and provides a variance estimator useful in practice.

A key aspect of this work is the choice of *representation* for elements of the Special Euclidean group $SE(3)$, each of which combines a translation in R^3 with a rotation in $SO(3)$. Ideally, we desire a representation that (1) supports vector addition and scaling, so that a principled noise model can be formulated, and (2) yields a simple form for g in Eq. 1, so that any singularities in the FIM can be readily identified.

We considered pairing translations with a number of rotation representations – Euler angles, Rodrigues parameters, and quaternions – but each lacks some of the criteria above. Instead, we compromise by representing each element of $SE(3)$ as a unit dual quaternion (DQ) in the space \mathbb{H} . Each DQ $q \in \mathbb{H}$ has eight parameters and can be expressed:

$$q = q_r + \varepsilon q_\varepsilon \quad (2)$$

where q_r is a “real” unit quaternion representing the rotation, q_ε is the “dual part” representing the translation, and $\varepsilon^2 = 0$. An 8-element DQ is “over-parametrized” (thus subject to two constraints) when representing a 6-DOF rigid body transform.

Although DQ’s are not minimal, they are convenient for this problem, combining in a way analogous to quaternion composition and yielding a simple form for g – about 20 lines of MatLab containing only polynomials and no trigonometric functions. An Euler-angle representation, by contrast, is minimal but produces much more complex expressions involving hundreds of lines of trigonometric terms. Homogeneous transformations yield a simple form of g , but require maintenance of many constraints. The DQ representation offers a good balance between compactness and convenience.

Ordinary additive Gaussian noise cannot be employed with DQ’s, since doing so would produce points not on $SE(3)$. Instead, we define a noise model using a projected Gaussian in the Lie algebra of DQ’s [9] which is appropriate for this over-parametrized form.

To identify singularities of the FIM, we adapt communication theory’s “blind channel estimation” methods to determine the calibration observability. Originally developed to determine the CRLB on constrained imaginary numbers [18], these methods extend naturally to DQ’s.

Background on calibration techniques and an introduction to DQ’s is provided in § II. The problem is formally stated in § III, along with a noise model appropriate for the DQ representation. System observability is proven, and degenerate cases are discussed, in § IV. The optimization process for constrained parameters is described in § V, with techniques for resampling asynchronous data and converting between representations provided in § VI. Experimental results from simulated and real data are given in § VII.

II. BACKGROUND

A. Calibration

There are several ways to determine the calibration. One can attempt to physically measure the rigid body transform between sensors. However, manual measurement can be made difficult by the need to establish an origin at an inaccessible location within a sensor, or to measure around solid parts of the body to which the sensors are mounted.

Alternatively, the calibration can be mechanically engineered through the use of precision machined mounts. This can work well for sensors in close proximity (e.g., stereo camera rigs), but is impractical for sensors placed far apart (e.g., on a vehicle or vessel).

The calibration could also be determined by establishing a global reference frame using Simultaneous Localization and Mapping (SLAM), then localizing each sensor within that frame (e.g., [11]). This approach has the significant disadvantage that it must invoke a robust SLAM solver as a subroutine.

Incremental motions have also been used to recover “hand-eye calibration” parameters. The authors in [5, 3, 20] recover the calibration between an end-effector and an adjacent camera by commanding the end-effector to move with some known velocity and estimating the camera motion. The degenerate conditions in [3, 20] are established through geometric arguments; we confirm their results via information theory and the CRLB. Further, the use of the CRLB allows our algorithm to identify a set of observations as degenerate, or nearly degenerate (resulting in a large variance), in practice. Dual quaternions were also used by [5]; we extend this notion and explicitly model the system noise via a projected Gaussian. Doing so allows us to confirm the CRLB in simulated experiments § VII-A.

B. Unit Dual Quaternions

Our calibration algorithm requires optimization over elements in $SE(3)$. Optimization over rigid body transformations is not new (e.g., [8]) and is a key component of many SLAM solutions. In our setting, unit dual quaternions (DQ’s) prove to be a convenient representation both because they yield a simple form for g (Eq. 1) and because they can be

implemented efficiently [14]. Optimization with DQ’s was also examined in [6], but their cost function included only translations; our optimization must simultaneously minimize rotation error. We review DQ’s briefly here; the interested reader should see [15, p. 53-62] for details.

A DQ q can be written in several ways: as an eight-element vector $[q_0, \dots, q_7]$; as two four-element vectors $[q_r, q_\varepsilon]$ (c.f. Eq. 2); or as a sum of imaginary and dual components:

$$q = (q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}) + \varepsilon(q_4 + q_5\mathbf{i} + q_6\mathbf{j} + q_7\mathbf{k}) \quad (3)$$

In this form, two DQ’s multiply according to the standard rules for the imaginary numbers $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$.

We write DQ multiplication as $a_1 \circ a_2$, where $\{a_1, a_2\} \in \mathbb{H}$. When we have vectors of DQ’s, e.g., $a = [a_1, a_2]$ and $b = [b_1, b_2]$, where $\{b_1, b_2\} \in \mathbb{H}$, we write $a \circ b$ to mean $[a_1 \circ b_1, a_2 \circ b_2]$.

A pure rotation defined by unit quaternion q_r is represented by the DQ $q = [q_r, 0, 0, 0, 0]$. A pure translation defined by $t = [t_0, t_1, t_2]$ can be represented by the DQ:

$$q = \left[1, 0, 0, 0, 0, \frac{t_0}{2}, \frac{t_1}{2}, \frac{t_2}{2} \right] \quad (4)$$

Given rigid body transform q , the inverse transform q^{-1} is:

$$q^{-1} = [q_0, -q_1, -q_2, -q_3, q_4, -q_5, -q_6, -q_7] \quad (5)$$

such that $q \circ q^{-1} = q^{-1} \circ q = \mathbf{I} = [1, 0, 0, 0, 0, 0, 0, 0]$. A vector $v = [v_0, v_1, v_2]$ can be represented as a DQ by:

$$q_v = [1, 0, 0, 0, 0, v_0, v_1, v_2] \quad (6)$$

The DQ form q_v of vector v transforms according to q as:

$$q'_v = q \circ q_v \circ q^* \quad (7)$$

where q^* is the dual-conjugate [14] to q :

$$q^* = [q_0, -q_1, -q_2, -q_3, -q_4, q_5, q_6, q_7] \quad (8)$$

DQ transforms can be composed as with unit quaternions; applying transform A , then transform B to point v yields:

$$q'_v = q_B \circ (q_A \circ q_v \circ q_A^*) \circ q_B^* = q_{BA} \circ q_v \circ q_{BA}^* \quad (9)$$

where $q_{BA} = q_B \circ q_A$. If the incremental motion v_{ri} and calibration k are expressed as DQ’s, then Eq. 1 becomes:

$$g(v_{ri}, k) := k^{-1} \circ v_{ri} \circ k \quad (10)$$

A DQ with $q_r^T q_r = 1$ and $q_r^T q_\varepsilon = 0$ (c.f. Eq. 2) has unit length. We impose these conditions as two constraints below.

C. Lie Groups & Lie Algebras

Lie groups are smooth manifolds for which associativity of multiplication holds, an identity exists, and the inverse is defined [7, 13]; examples include R^n , $SO(3)$, and $SE(3)$. However, R^n is also a vector space (allowing addition, scaling, and commutativity), but $SO(3)$ and $SE(3)$ are not. For this reason, points in $SO(3)$ and $SE(3)$ cannot simply be interpolated or averaged. We use the Lie algebra to enable an optimization method that requires these operations.

Lie algebra describes a local neighborhood (i.e., a tangent space) of a Lie group. So the Lie algebra of DQ’s, \mathfrak{h} , can be used to express a vector space tangent to some point in \mathbb{H} . Within this vector space, DQ’s can be arithmetically

manipulated. Once the operation is complete, points in the Lie algebra can be projected back into the Lie group.

The logarithm maps from the Lie group to the Lie algebra; the exponent maps from the Lie algebra to the Lie group. Both mappings are done at the identity. For example, if we have two elements of a Lie group $\{u, v\} \in \mathbb{H}$, the ‘‘box’’ notation of [9] expresses the difference between v and u as $d = v \boxminus u$. (Here, $\{u, v\}$ are each eight-element vectors and d is a six-element vector.) That is, the box-minus operator connotes $d = 2 \log(u^{-1} \circ v)$ where $d \in \mathfrak{h}$. In the Lie group, $u^{-1} \circ v$ is a small transformation relative to the identity, i.e., relative to $u^{-1} \circ u$ (see Fig. 2). The factor of two before the log is a result of the fact that DQ’s are multiplied on the left and right of the vector during transformation (c.f. Eq. 7).

Similarly, the box-plus addition operator [9] involves exponentiation. If $d \in \mathfrak{h}$, then $\exp d \in \mathbb{H}$. If $d = 2 \log(u^{-1} \circ v)$, then $u \circ \exp \frac{d}{2} = u \circ \exp(\log(u^{-1} \circ v)) = u \circ u^{-1} \circ v = v$. Since d applies a small transform to u , we use the box-plus operator to write $v = u \boxplus d = u \circ \exp \frac{d}{2}$.

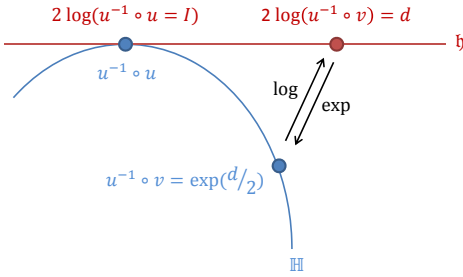


Fig. 2: The mapping between the Lie group, \mathbb{H} , and the Lie algebra, \mathfrak{h} , is performed at the identity, $u^{-1} \circ u$.

This definition of the difference between DQ’s yields a Gaussian distribution as follows: imagine a Gaussian drawn on the line \mathfrak{h} in Fig. 2. Exponentiating points on this Gaussian ‘‘projects’’ the distribution onto \mathbb{H} . This projected Gaussian serves as our noise model (§ III).

Summarizing, the Lie group/algebra enables several key operations: (1) addition of two DQ’s, (2) subtraction of two DQ’s, and (3) addition of noise to a DQ.

1) *Logarithm of dual quaternions:* The logarithm of some $q \in \mathbb{H}$ can be calculated as [17]:

$$\begin{aligned} \log q = & \left(\frac{1}{4(\sin \theta)^3} [(2\theta - \sin(2\theta))q^3 \right. \\ & + (-6\theta \cos \theta + 2 \sin(3\theta))q^2 \\ & + (6\theta - \sin(2\theta) - \sin(4\theta))q \\ & \left. + (-3\theta \cos \theta + \theta \cos(3\theta) \right. \\ & \left. - \sin \theta + \sin(3\theta))\mathbf{I} \right]_{1:3,5:7} \quad (11) \end{aligned}$$

where θ is the rotation angle associated with the DQ, and exponentiation of a DQ is implemented through repeated multiplication (\circ). (This expression incorporates a correction, provided by Selig, to that given in [17].) The $(\cdot)_{1:3,5:7}$ removes the identically zero-valued first and fifth elements from the 8-vector. To avoid the singularity at $\theta = 0$, the limit of $\log q$ can

be evaluated as:

$$\lim_{\theta \rightarrow 0} \log q = [0, 0, 0, q_5, q_6, q_7] \quad (12)$$

For compactness, we write $\log a$ to mean $[\log a_1, \log a_2]$.

2) *Exponential of dual quaternions:* If $d \in \mathfrak{h}$, $w = [0, d_0, d_1, d_2]$ is a quaternion and $q = [w, 0, d_3, d_4, d_5]$, we can exponentiate d as [17]:

$$\begin{aligned} \exp d = & \frac{1}{2}(2 \cos |w| + |w| \sin |w|)\mathbf{I} \\ & - \frac{1}{2}(\cos |w| - 3 \operatorname{sinc} |w|)q + \frac{1}{2}(\operatorname{sinc} |w|)q^2 \\ & - \frac{1}{2|w|^2}(\cos |w| - \operatorname{sinc} |w|)q^3 \quad (13) \end{aligned}$$

The singularity at $w = 0$ can be avoided by evaluating:

$$\lim_{|w| \rightarrow 0} \exp d = \mathbf{I} + q \quad (14)$$

III. PROBLEM STATEMENT

Given the observed incremental motions and their covariances, the problem is to estimate the most likely calibration. We formulate this task as a non-linear least squares optimization with a state representation of DQ’s. DQ’s were chosen to verify the CRLB in § VII-A. In § V, we show how to perform the calculation in the over-parametrized state space. (We chose an over-parametrization rather than a minimal representation, in order to avoid singularities.)

The calibration to estimate is $k = [k_0, \dots, k_7]$, where $k \in \mathbb{H}$. The 6-DOF incremental poses from each sensor form a series of DQ observations (our experiments use FOVIS [10] and KinectFusion [16]). Let $z_r = [z_{r1}, z_{r2}, \dots, z_{rN}]$ and $z_s = [z_{s1}, z_{s2}, \dots, z_{sN}]$ be the observations from sensor r and s , respectively. Note that $\{z_{ri}, z_{si}\} \in \mathbb{H}$. Finally, let $z = [z_r, z_s]$ be the $(2N)$ observations. Both the incremental poses of sensor r and the calibration must be estimated [2], so the state to estimate is $x = [v_r, k]$ consisting of $(N + 1)$ DQ’s, where $v_r = [v_{r1}, v_{r2}, \dots, v_{rN}]$ is the latent incremental motion of sensor r .

We then formulate the task as a maximum likelihood optimization [2]:

$$\hat{x}^{ML}(z) = \operatorname{argmax}_{x=[v_r, k]} \prod_{i=1}^N P(z_{ri}|v_{ri})P(z_{si}|v_{ri}, k) \quad (15)$$

under the constraint that $\{v_{ri}, k\} \in \mathbb{H}$.

The probability functions might be assumed to be Gaussian. However, it is clear that adding Gaussian noise to each term of a DQ will not generally result in a DQ. Instead, we use the projected Gaussian.

By comparison, other approaches [5, 3, 20] simply ignore noise and assume that observations and dimensions should be equally weighted. However, when observation uncertainty varies (e.g., when the number of features varies for a vision algorithm) or when the uncertainty among dimensions varies (e.g., translations are less accurate than rotations), explicit representation of noise minimizes the effects of inaccurate observations. Further, a principled noise model allows recovery of the CRLB and, thus, the calibration’s uncertainty.

A. Process model

The process model can be written as:

$$z = G(x) \circ \exp \frac{\delta}{2} \equiv G(x) \boxplus \delta \quad (16)$$

$$G(x) = [v_{r1}, \dots, v_{rN}, g(v_{r1}, k), \dots, g(v_{rN}, k)] \quad (17)$$

where $\delta \in \mathfrak{h}$ acts as a projected Gaussian: $\delta \sim N(0, \Sigma_z)$. Here, the expected observations $G(x)$ have been corrupted by a noisy transformation with δ . Notice that the process model is not typical additive Gaussian noise, $z = G(x) + \delta$, which would result in $z \notin \mathbb{H}$.

The difference between the observations, z , and the expected values, $G(x)$, is $\lambda = -2 \log(z^{-1} \circ G(x))$, where λ includes $12N$ parameters, six for each of the $2N$ observations. The posteriors in Eq. 15 can be written:

$$P(z|x) \sim f(z^{-1} \circ G(x))$$

$$f(z^{-1} \circ G(x)) = \frac{1}{\sqrt{(2\pi)^{12N} |\Sigma_z|}} e^{-\frac{1}{2} \lambda^T \Sigma_z^{-1} \lambda} \quad (18)$$

IV. OBSERVABILITY

Assume we have some estimate \hat{x} of the true parameters x_0 . We wish to know how the estimate varies, so we calculate the covariance $E[(\hat{x} - x_0)(\hat{x} - x_0)^T]$. Cramer and Rao [19] showed that this quantity can be no smaller than the inverse of J , the Fisher Information Matrix (FIM):

$$E[(\hat{x} - x_0)(\hat{x} - x_0)^T] \geq J^{-1} \quad (19)$$

The CRLB is critical because (1) it defines a best case covariance for our estimate, and (2) if J is singular, no estimate exists. If (and only if) x cannot be estimated, then x is unobservable. Indeed, we wish to identify the situations under which x is unobservable and arrange to avoid them in practice.

A. Shift invariance

For an unbiased estimator, the FIM is:

$$J = E[(\nabla_x \ln P(z|x))(\nabla_x \ln P(z|x))^T] \quad (20)$$

The FIM, as it is based on the gradient, is invariant under rigid body transformations, so the FIM of a distribution of $f(z^{-1} \circ G(x))$ equals that of the distribution $f(G(x))$. This is because the expectation is an integration over $SE(3)$ and is unaffected by a shift [4]. Thus, we analyze $\bar{P}(z|x) \sim f(G(x))$ to produce an equivalent FIM.

Let $H(y) = -2 \log y$ and $\bar{\lambda} = H(G(x))$. H accepts a $16N \times 1$ parameter vector of DQ's in the Lie group, and returns a $12N \times 1$ vector of differences in the Lie algebra.

B. Fisher Information Matrix

In order to calculate the FIM for our estimation problem, we first find the gradient of $\bar{\lambda}$ by applying the chain rule:

$$\begin{aligned} \nabla_x \bar{\lambda} &= [\nabla_p H(p)|_{p=G(x)}]^T [\nabla_x G(x)]^T \\ &= \underbrace{J_H}_{12N \times 16N} \underbrace{J_G}_{16N \times 8(N+1)} \end{aligned} \quad (21)$$

Then, we calculate:

$$\nabla_x \ln P(z|x) = \nabla_x \ln c e^{-\frac{1}{2} \bar{\lambda}^T \Sigma_z^{-1} \bar{\lambda}} \quad (22)$$

$$= \nabla_x \left(-\frac{1}{2} \bar{\lambda}^T \Sigma_z^{-1} \bar{\lambda} \right) \quad (23)$$

$$= (J_H J_G)^T \Sigma_z^{-1} \bar{\lambda} \quad (24)$$

$$= J_G^T J_H^T \underbrace{\Sigma_z^{-1}}_{12N \times 12N} \underbrace{\bar{\lambda}}_{12N \times 1} \quad (25)$$

Substituting into Eq. 20:

$$J = E[(\nabla_x \ln P(z|x))(\nabla_x \ln P(z|x))^T] \quad (26)$$

$$= E[(J_G^T J_H^T \Sigma_z^{-1} \bar{\lambda})(J_G^T J_H^T \Sigma_z^{-1} \bar{\lambda})^T] \quad (27)$$

$$= E[J_G^T J_H^T \Sigma_z^{-1} \bar{\lambda} \bar{\lambda}^T (\Sigma_z^{-1})^T J_H J_G] \quad (28)$$

$$= J_G^T J_H^T \Sigma_z^{-1} E[\bar{\lambda} \bar{\lambda}^T] (\Sigma_z^{-1})^T J_H J_G \quad (29)$$

$$= J_G^T J_H^T \Sigma_z^{-1} \Sigma_z (\Sigma_z^{-1})^T J_H J_G \quad (30)$$

$$= J_G^T J_H^T \Sigma_z^{-1} J_H J_G \quad (31)$$

Since each of the $(N+1)$ DQ's in x is subject to two constraints, J_G is always rank-deficient by at least $2(N+1)$. Of course, our interest is not in rank deficiencies caused by over-parametrization but in singularities due to the observations. We must distinguish singularities caused by over-parametrization from those caused by insufficient or degenerate data.

C. Cramer-Rao Lower Bound

In the communications field, a related problem is to estimate complex parameters of the wireless transmission path. These complex "channel" parameters are usually unknown but obey some constraints (e.g., they have unit magnitude). Since the CRLB is often useful for system tuning, Stoica [18] developed techniques to determine the CRLB with constrained parameters.

Following [18], suppose the m constraints are expressed as $f^c = [f_1^c, \dots, f_m^c]$ such that $f^c(x) = 0$. Let F^c be the Jacobian of f^c and U be the null space of F^c such that $F^c(x)U = 0$. When $K = U^T J U$ is non-singular, the CRLB exists. In our case,

$$K = U^T J U = U^T J_G^T J_H^T \Sigma_z^{-1} J_H J_G U \quad (32)$$

$$= U^T J_G^T J_H^T L L^T J_H J_G U \quad (33)$$

$$= (L^T J_H J_G U)^T (L^T J_H J_G U) \quad (34)$$

where $\Sigma_z^{-1} = L L^T$ by the Cholesky factorization. In order to find the cases where J is singular, we examine the rank of K :

$$\text{rank}(K) = \text{rank}((L^T J_H J_G U)^T (L^T J_H J_G U)) \quad (35)$$

Since $\text{rank}(A^T A) = \text{rank}(A)$,

$$\text{rank}(K) = \text{rank}(L^T J_H J_G U) \quad (36)$$

Further, since each observation is full rank, Σ_z is full rank; L^T is a $12N \times 12N$ matrix with rank $12N$ and $\text{rank}(L^T A) = \text{rank}(A)$. Thus (see Appendix A):

$$\text{rank}(K) = \text{rank}(J_H J_G U) = \text{rank}(J_G U). \quad (37)$$

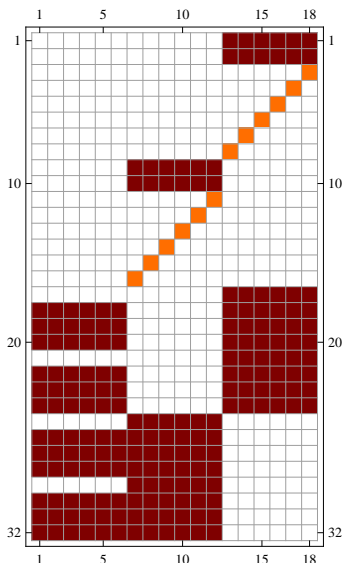


Fig. 3: Visualization of the matrix $J_G U$ shows that only the first six columns can reduce. Blank entries are zero; orange are unity; red are more complex quantities. (Full expressions for the matrix elements shown in red are included at the URL given in § I.)

D. Degeneracies

As shown in [2], one observation is insufficient to recover the calibration. For two observations and a particular null space matrix, U , $J_G U$ has the form shown in Fig. 3. Notice that columns 7-18 are always linearly independent due to the unity entries. This is not surprising since v_{ri} , the estimated motion of sensor r , is directly observed by z_{ri} . Only the first six columns, corresponding to the six DOF's of the calibration, can possibly reduce. By examining linear combinations of these columns, we can identify a singularity which, if avoided in practice, will ensure that the calibration is observable.

Suppose the two incremental motions experienced by sensor r are $\{a, b\} \in \mathbb{H}$ and let a_i be the i -th term of the 8-element DQ, a . When $a_1 b_3 = a_3 b_1$ and $a_2 b_3 = a_3 b_2$, $J_G U$ is singular and the calibration is unobservable. Since the 2nd-4th elements of the DQ correspond to the 2nd-4th elements of a unit quaternion representing the rotation, it follows that these relations hold only when there is no rotation or when the rotation axes of a and b are parallel. Thus, the *calibration is unobservable when the sensors rotate only about parallel axes*.

In principle, this analysis could have been completed using any representation for $SE(3)$. However, attempting the analysis using Euler angles and Mathematica 8.0 exceeded 24GB of memory without completing; manual analysis was equally difficult. By contrast, DQ over-parametrization made both manual and automated analyses tractable to perform and simple to interpret, making readily apparent the degeneracy arising from a fixed axis of rotation.

The condition to avoid degeneracy has several common special cases:

- 1) Constant velocity. When the sensors move with constant velocity, the axes of rotation are constant.
- 2) Translation only. When the sensors only translate, no rotation is experienced.
- 3) Planar motion. When the robot travels only in a plane, the rotation axis is fixed (perpendicular to the plane).

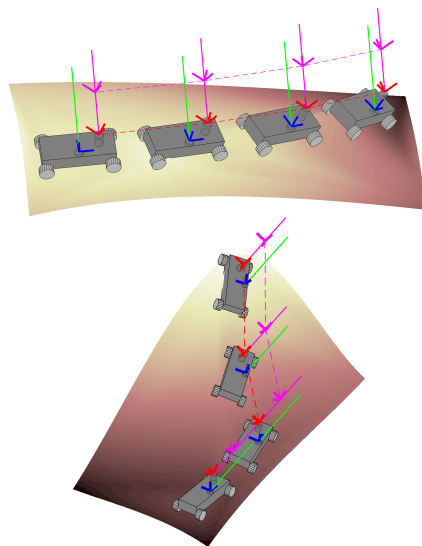


Fig. 4: Two robots driven along the suggested paths experience rotation about only one axis (green). As a result, the true calibration relating the two true sensor frames (red/blue) cannot be determined. The magenta lines and frames show ambiguous locations for the red sensor frame.

These special cases, however, do not fully characterize the degeneracy. So long as the axis of rotation of the incremental poses remains fixed, any translations and any magnitude of rotation will not avoid singularity.

In Fig. 4, for example, a robot is traveling along the suggested terrain. Although the robot translates and rotates some varying amount between poses, it always rotates about the same axis (green lines). In such situations, the calibration is ambiguous at least along a line parallel to the axis of rotation (magenta line). That is, if the calibration is translated along such a line, the observations from sensor s remain fixed. Thus, because multiple calibrations result in the same observations, the calibration is unobservable.

V. OPTIMIZATION

In order to estimate the calibration in Eq. 15, we perform non-linear least squares optimization, using a modified Levenberg-Marquardt (L-M) algorithm [9]. The optimization proceeds as:

$$x_{t+1} = x_t \boxplus - (\mathcal{J}_t^T \Sigma^{-1} \mathcal{J}_t)^{-1} \mathcal{J}_t^T \Sigma^{-1} (G(x_t) \boxminus z) \quad (38)$$

The term $G(x_t) \boxminus z$ represents error elements in \mathfrak{h} , which are scaled by the gradient and added (via \boxplus) to the current parameter estimate to produce a new set of DQ's. The method computes error, and applies corrections to the current estimates, in the tangent space via the Lie algebra. After each update, the parameters lie in \mathbb{H} .

\mathcal{J}_t is the analytic Jacobian at the current estimate [21]:

$$\mathcal{J}_t = \left[\nabla_h H \left(z^{-1} \circ G \left(x_t \circ \exp \left(\frac{h}{2} \right) \right) \right) \right] \Big|_{h=0} \quad (39)$$

Essentially, the method shifts the parameters x_t via $h \in \mathfrak{h}$, then evaluates that shift at $h = 0$.

VI. INTERPOLATION

Although the DQ representation facilitates the FIM analysis, and there are methods to develop a noise model, data and covariance matrices will typically be available in more common formats, such as Euler angles. Furthermore, sensors are rarely synchronized, so incremental motions may be observed over different sample periods. Following the process in [2], we use the Scaled Unscented Transform (SUT) [12] to (1) convert incremental motion and covariance data to the DQ representation and (2) resample data from different sensors at common times.

The SUT creates a set of sigma points, \mathcal{X} , centered about the mean and spaced according to the covariance matrix. Each point is passed through the interpolation function f^i to produce a set of transformed points, \mathcal{Y} . A new distribution is then created to approximate the weighted \mathcal{Y} . We employ the process adapted by [9] to incorporate the Lie algebra.

First, f^i converts the Euler states to DQ's. Second, it accumulates the incremental motions into a common reference frame and resamples them at the desired instants, typically the sample times of the reference sensor. This resampling is done via the DQ SLERP operator [14] which interpolates between poses with constant speed and shortest path. The function then calculates the incremental motions.

The SUT requires that transformed sigma points, \mathcal{Y} , be averaged according to some weights, w . Because averaging each individual element of a DQ would not result in a DQ, the mean of a set of DQ's must be estimated by optimization. We wish to find the mean \hat{b} that minimizes the error term in:

$$\hat{b}(w, \mathcal{Y}) = \operatorname{argmin}_b \sum_{i=1}^N w_i \log(b^{-1} \circ \mathcal{Y}_i) \quad (40)$$

The averaging procedure given in [9], intended for non-negative weights, fails to converge when the \mathcal{Y}_i 's are similar (i.e., for small covariance). Since our SUT implementation can return negative weights, we use L-M optimization (Eq. 38) with the mean cost function given in Eq. 40.

VII. RESULTS

A. Simulation

We validated the calibration estimates and constrained CRLB by simulating robot motion along the path shown in Fig. 5. The $N = 62$ observations and covariances per sensor were simulated using a translation/Euler angle parametrization (instead of DQ's) to model data obtained from sensors in practice. Additive Gaussian noise is applied to this minimal representation with a magnitude 10-40% of the true value.

Thirty different calibrations were simulated, each with rigid body parameters k uniformly drawn from $[\pm 3 \text{ m}, \pm 3 \text{ m}, \pm 3 \text{ m}, \pm \pi, \pm \pi, \pm \pi]$. The observations and covariances were then converted to DQ's using the interpolation method of § VI. We validated the CRLB by performing 400 Monte Carlo simulations [1] for each calibration, sampling each velocity vector from its Gaussian distribution.

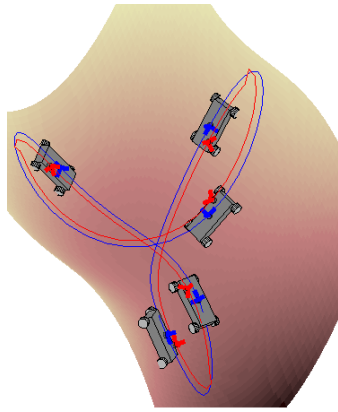


Fig. 5: Motion simulated such that the red and blue sensors traveled the paths shown. The path is always non-degenerate; in this image $k = [0.1, 0.05, 0.01, 0, 0, \frac{\pi}{3}]$.

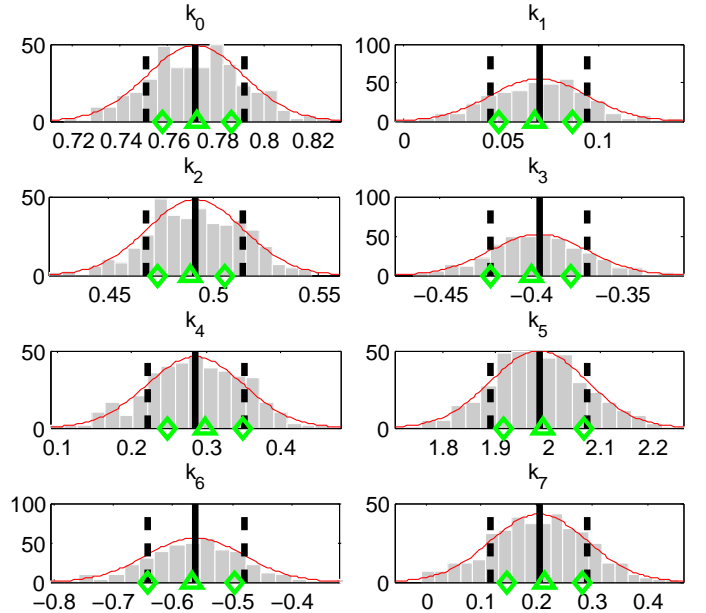


Fig. 6: Histograms (gray) of calibration estimates from 400 simulations of the path in Fig. 5 match well with the true calibration (green triangles) and constrained CRLB (green diamonds). Black lines indicate the sample mean (solid) and one standard deviation (dashed); red lines show a fitted Gaussian.

Fig. 6 shows results for a sample calibration:

$k_{Euler} = [2.79 \text{ m}, -2.79 \text{ m}, -1.45 \text{ m}, -0.51 \text{ r}, 0.94 \text{ r}, -1.22 \text{ r}]$
in meters (m) and radians (r) or, in DQ form, at:

$$k_{DQ} = [0.77, 0.07, 0.49, -0.40, 0.30, 1.99, -0.57, 0.21]$$

As shown, the mean and standard deviations from the simulations match well with the true value and the CRLB, respectively. It is important to note that the covariance matrix corresponding to Fig. 6 is calculated on an over-parametrization; there are only six DOF's in the 8-element DQ representation. Due to these dependent (i.e., constrained) parameters, the covariance matrix is singular. However, because we use the Lie algebra during optimization and filtering, we can employ the DQ parametrization to avoid problems associated with singular covariances.

Fig. 7 shows the error between the thirty true calibrations and the mean of the estimated values for one of the DQ

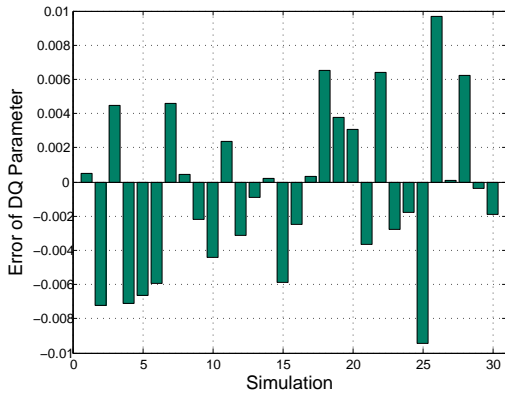


Fig. 7: The error between the known calibration and the mean estimate was less than ± 0.01 for each DQ parameter.

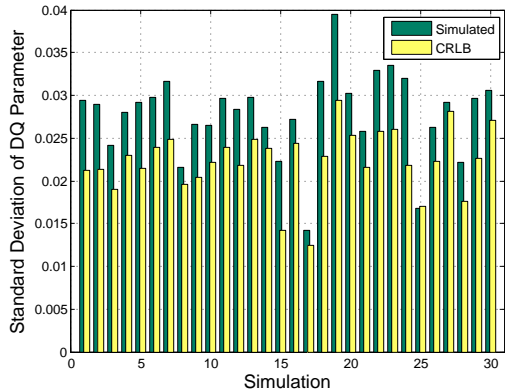


Fig. 8: The standard deviation of the simulations and predicted CRLB agreed to within ~ 0.01 for each DQ parameter.

parameters. The other DQ elements (not shown) behaved similarly; they were recovered to within about 0.01 of truth. Fig. 8 compares the standard deviation of the parameters resulting from the Monte Carlo experiments and the predicted CRLB. In general, the method came close (within about 0.01) to the best-case CRLB.

B. Real data

We further validated the estimator with two different types of depth sensors and motion estimation algorithms. First, we collected data with two Microsoft Kinect RGB-D cameras, mounted on three different machined rigs with known calibrations. The RGB-D data from each camera was processed using the Fast Odometry from VISion (FOVIS) [10] library, which uses image features and depth data to produce incremental motion estimates. Second, we collected data with two rigidly mounted Xtion RGB-D cameras and used the KinectFusion algorithm [16] for motion estimation. For all four calibrations, we moved each rig by hand along a path in 3D. The interpolation algorithm (§ VI) was used to synchronize the translations/Euler angles and convert to DQs.

We characterized the noise in both systems using data from stationary sensors. We projected the noise into \mathfrak{h} and, using a chi-squared goodness-of-fit test, found the projected Gaussian to be a good approximation (at 5% significance) for both FOVIS and KinectFusion.

TABLE I: Calibrations recovered from real data

	x (m)	y (m)	z (m)	ρ (r)	ϑ (r)	ψ (r)
True 1 ^a	-0.045	-0.305	-0.572	-1.316	0.906	-1.703
Error ^b	0.000	0.011	0.011	0.022	0.002	0.009
True 2 ^a	-0.423	-0.004	0.006	-0.000	0.000	3.141
Error ^b	-0.007	-0.014	-0.003	-0.019	0.003	-0.007
True 3 ^a	-0.165	0.204	-0.244	1.316	-0.906	3.009
Error ^b	-0.007	0.003	-0.013	0.003	0.000	-0.005
True 4 ^a	-0.040	0.025	0.000	-0.052	0.000	3.141
Error ^b	-0.006	0.008	0.006	0.013	-0.017	0.005

^a Ground truth calibration

^b Difference between mean of the estimates and the true calibration

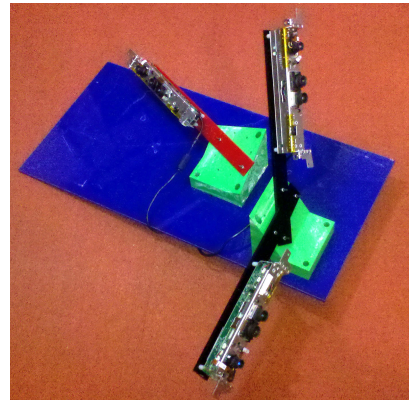


Fig. 9: We assess the method’s consistency by recovering the loop of calibrations relating three RGB-D sensors.

Table I summarizes the results of the four different calibrations. For clarity, the transforms are shown as translations and Euler angles, but all processing was done with DQ’s. We assumed a constant variance for each DOF. The first three calibrations used Kinects and FOVIS with $2N \simeq 2000$ observations; the last used Xtions and KinectFusion with $2N \simeq 400$. In each case, the method recovered the inter-sensor translation to within about 1.4 cm, and the rotation to within about 1.26 degrees.

We assessed the estimator’s consistency with three rigidly mounted depth cameras r , s , and t (Fig. 9). We estimated the pairwise calibrations $k_{s,r}$, $k_{t,s}$, and $k_{r,t}$, where, e.g., $k_{s,r}$ is the calibration between r and s . The closed loop of estimated transformations should return to the starting sensor frame:

$$k_{s,r} \circ k_{t,s} \circ k_{r,t} = \mathbf{I} \quad (41)$$

The accumulated error was small: translation error was $[-4.27, -2.66, 7.13]$ mm and rotation error (again in Euler angles) was $[7.03, -5.20, -1.49]$ milliradians.

VIII. CONCLUSION

We described a practical method that recovers the 6-DOF rigid body transform between two sensors, from each sensor’s observations of its 6-DOF incremental motion. Our contributions include treating observation noise in a principled manner, allowing calculation of a lower bound on the uncertainty of the

estimated calibration. We show that the system is unobservable when rotation occurs only about parallel axes.

Additionally, we illustrate the use of a constrained DQ parametrization which greatly simplified the algebraic machinery of degeneracy analysis. Such over-parametrizations are typically avoided in practice, however, because they make it difficult to perform vector operations (addition, scaling, averaging, etc.), develop noise models, and identify system singularities. We assemble the tools for each required operation, employing the Lie algebra to define local vector operations and a suitable projected Gaussian noise model. Finally, we demonstrated that the constrained form of the CRLB enables system observability to be shown.

The method accurately recovers the 6-DOF transformations relating pairs of asynchronous, rigidly attached sensors, requiring only hand-held motion of the sensors through space.

We gratefully acknowledge the support of the Office of Naval Research through award #N000141210071.

APPENDIX

If A is $D \times E$, B is $E \times F$, $\mathcal{N}(A)$ is the null space of A , $\mathcal{R}(B)$ is the column space of B , and $\dim A$ is the number of vectors in the basis of A , then $\text{rank}(AB) = \text{rank}(B) - \dim[\mathcal{N}(A) \cap \mathcal{R}(B)]$. Substituting from Eq. 37,

$$\text{rank}(J_H J_G U) = \text{rank}(J_G U) - \dim[\mathcal{N}(J_H) \cap \mathcal{R}(J_G U)]$$

Intuitively, this means that if a column of $J_G U$ lies in the null space of J_H , information is lost during the multiplication and the rank of the matrix product is reduced. In order to show that $\text{rank}(J_H J_G U) = \text{rank}(J_G U)$, there are two cases:

- 1) If $J_G U$ is singular, then $\text{rank}(J_G U) < 6(N + 1)$, where N is the number of observations. This implies $\text{rank}(J_H J_G U) < 6(N + 1)$. Thus, $J_G U$ is singular implies $J_H J_G U$ is singular.
- 2) If $J_H J_G U$ is singular, then either $J_G U$ is singular or $\dim[\mathcal{N}(J_H) \cap \mathcal{R}(J_G U)] > 0$.
 - If $J_G U$ is singular, then this is the case above.
 - If $J_G U$ is not singular, then $\text{rank}(J_G U) = 6(N + 1)$. The task then becomes to determine $\dim[\mathcal{N}(J_H) \cap \mathcal{R}(J_G U)]$. Since $J_G U$ is full rank, $\mathcal{R}(J_G U)$ is the columns of $J_G U$. Furthermore, there are $4N$ columns in $\mathcal{N}(J_H)$, one for each of the two constraints of the $2N$ DQ's. (Fig. 10 shows $\mathcal{N}(J_H)$ for $N = 2$.) It can be shown that $\text{rank}([J_H, J_G U]) = \text{rank}(J_H) + \text{rank}(J_G U)$. In other words, none of the columns of $\mathcal{N}(J_H)$ will intersect with the columns of $J_G U$. Thus, $\mathcal{N}(J_H) \cap \mathcal{R}(J_G U) = \emptyset$ and $\text{rank}(J_H J_G U) = \text{rank}(J_G U)$. Since $J_G U$ is not singular, $J_H J_G U$ is not singular, which is a contradiction. Only the former possibility remains, and $J_H J_G U$ is singular implies $J_G U$ is singular.

In conclusion, $J_H J_G U$ is singular if and only if $J_G U$ is singular. Intuitively, this is not a surprising result; the log function is designed to preserve information when mapping between the Lie group and the Lie algebra.

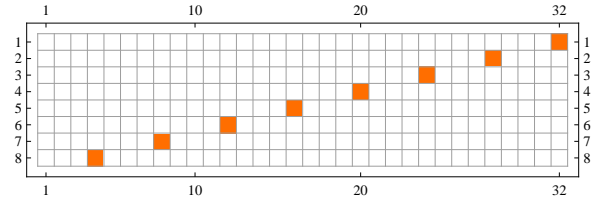


Fig. 10: The matrix $\mathcal{N}(J_H)^T$, depicted here for $N = 2$, reveals $4N$ DOF's corresponding to the constraints of the $2N$ DQ's in z . Blank entries are zero; orange are unity.

REFERENCES

- [1] Y. Bar-Shalom, T. Kirubarajan, and X. Li. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2002.
- [2] J. Brookshire and S. Teller. Automatic calibration of multiple coplanar sensors. *RSS*, 2011.
- [3] H. H. Chen. A screw motion approach to uniqueness analysis of head-eye geometry. In *CVPR*, Jun 1991.
- [4] G. Chirikjian. Information theory on Lie groups and mobile robotics applications. In *ICRA*, 2010.
- [5] K. Daniilidis. Hand-eye calibration using dual quaternions. *IJRR*, 18, 1998.
- [6] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Mach. Vision Appl.*, 9(5-6), March 1997.
- [7] V. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *CVPR*, 2004.
- [8] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3D. In *IROS*, Nov 2007.
- [9] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *CoRR*, 2011.
- [10] A. Huang, A. Bachrach, P. Henry, et al. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *ISSR*, Aug 2011.
- [11] E. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *IJRR*, Oct 2010.
- [12] S. Julier. The scaled unscented transformation. In *Proc. ACC*, volume 6, pages 4555–4559, 2002.
- [13] K. Kanatani. *Group Theoretical Methods in Image Understanding*. Springer-Verlag New York, Inc., 1990.
- [14] L. Kavan, S. Collins, J. Zara, and C. O'Sullivan. Geometric skinning with approximate dual quaternion blending. volume 27. ACM Press, 2008.
- [15] J. McCarthy. *An Introduction to Theoretical Kinematics*. MIT Press, 1990.
- [16] R. Newcombe, A. Davison, S. Izadi, et al. KinectFusion: real-time dense surface mapping and tracking. In *ISMAR*, Oct. 2011.
- [17] J. Selig. Exponential and Cayley maps for dual quaternions. *Adv. in App. Clifford Algebras*, 20, 2010.
- [18] P. Stoica and B. C. Ng. On the Cramer-Rao bound under parametric constraints. *Signal Processing Letters, IEEE*, 5(7):177–179, Jul 1998.
- [19] H. Van Trees. *Detection, Estimation, and Modulation Theory, Part I*. John Wiley & Sons, New York, 1968.
- [20] R. Y. Tsai and R. K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans. Robot. Autom.*, 5(3), Jun 1989.
- [21] A. Ude. Nonlinear least squares optimisation of unit quaternion functions for pose estimation from corresponding features. In *ICPR*, volume 1, Aug 1998.