A Probabilistic Solution to the AX=XB Problem: Sensor Calibration without Correspondence

M. Kendal Ackerman and Gregory S. Chirikjian*

Dept. of Mechanical Eng. - Johns Hopkins University - Baltimore, MD, USA
gregc@jhu.edu

Abstract. The "AX=XB" sensor calibration problem is ubiquitous in the fields of robotics and computer vision. In this problem A, X, and B are each homogeneous transformations (i.e., rigid-body motions) with A and B given from sensor measurements, and X is the unknown that is sought. For decades this problem is known to be solvable for X when a set of exactly measured compatible A's and B's with known correspondence is given. However, in practical problems, it is often the case that the data streams containing the A's and B's will present at different sample rates, they will be asynchronous, and each stream may contain gaps in information. Practical scenarios in which this can happen include hand-eye calibration and ultrasound image registration. We therefore present a method for calculating the calibration transformation, X, that works for data without any a priori knowledge of the correspondence between the As and Bs.

Keywords: $AX = XB$, Sensor Calibration, Information Fusion, Probabilistic Modeling.

1 Introduction

The "AX=XB" sensor calibration problem is well-known in the fields of robotics and computer vision. In this problem A, X, and B are each homogeneous transformations (i.e., elements of the special Euclidean group, $SE(3)$) with A and B given from sensor measurements, and X unknown.

This paper presents a method to solve for an X wherein there does not need to be any a priori knowledge of the correspondence between A's and B's. In other words, the sets of A's and B's each can be given as unordered "batches". This method hinges on viewing each of the A's, B's and X as shifted Dirac delta functions on the group of $SE(3)$. Since functions can be added, it is possible to write a single, "correspondence-free", equation relating functions of the A's and B's with a convolved delta function in X. Additionally, it is often the case that the A's and B's are small relative motions (due to high sensor sampling rates), creating "highly focused" distributions with useful properties. This facilitates solving a set of equations that relate the means and covariances of the A and B distributions.

* This work was supported by NSF Grant RI-Medium: IIS-1162095

F. Nielsen and F. Barbaresco (Eds.): GSI 2013, LNCS 8085, pp. 693–701, 2013.

The remainder of this section is devoted to reviewing the literature, and establishing notation. Section 2 presents the theory for our new batch solution method. Section 3 shows the results of the batch solution method for simulated data.

1.1 Literature Review

Any (proper) rigid-body motion in three-dimensional space can be described as a 4×4 homogeneous transformation matrix of the form

$$H(R, \mathbf{t}) = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \tag{1}$$

where $R \in SO(3)$ is a 3×3 (proper) rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector. The set of all such matrices can be identified with $SE(3)$, the group of rigid-body motions, where the group law is matrix multiplication. The emphasis in this paper is solving the problem

$$AX = XB \tag{2}$$

for $X \in SE(3)$ when multiple pairs of $(A, B) \in SE(3) \times SE(3)$ are presented, but when the correspondence between them has been lost. The version of the problem with known correspondence has a history that goes back more than a quarter of a century [2]-[4], and applications involving this problem remain active today [5]-[7].

It is well known that, in non-degenerate cases, there are two unspecified degrees of freedom to the problem for a single pair of sensor measurements, (A, B). This situation is rectified by considering two pairs of exact measurements of the form in (2), i.e., $A_1X = XB_1$ and $A_2X = XB_2$, provided that some mild conditions are observed for the selection of the pairs (A_1, B_1) and (A_2, B_2) [1, 3, 4]. Additionally, if there is sensor error, it may not be possible to find compatible pairs that reproduce the exact value of X. For this reason, minimization and least squared approaches often take large sets of A's and B's.

Moreover, traditional solution procedures assume that there is exact knowledge of the A_i and B_i correspondence, which is not always the case. There are many instances in the literature when the sensor data used in calibration becomes "unsynchronized". For example, an image guided therapy (IGT) system may have an ultrasound (US) transducer with attached optical marker or electromagnetic (EM) tracker. These individual data streams will likely be unsynchronized because data is often gathered on distributed networks [9]. In addition to sampling rate and temporal alignment issues, data can often be dropped or misread due to transient occlusions of the sensors, creating gaps in the data streams. In addition to US and hand-eye calibration, another of the many examples of sensor stream gaps occurs in the calibration of aerial vehicle IMU and camera data [10].

Different attempts have been implemented to solve this problem, such as time stamping the data, developing dedicated software modules for syncing the data [9], and analyzing components of the sensor data stream to determine a correlation [10], to varying effects. However, as systems become more distributed, a common clock may not be present, and methods that require explicit knowledge of correspondence are not designed for this scenario. The new solution methodology presented here bypasses these issues altogether and solves the problem without tracking, or recomputing, correspondence.

1.2 Notation and Mathematical Problem Formulation

Given a large set of pairs $(A_i, B_i) \in SE(3) \times SE(3)$ for $i = 1, ..., n$ that exactly satisfy the equation

$$A_i X = X B_i \tag{3}$$

numerous algorithms exist to find $X \in SE(3)$, as discussed earlier. Here we address a generalization of this problem in which the sets $\{A_i\}$ and $\{B_j\}$ are provided with elements written in any order and it is known that a correspondence exists between the elements of these sets such that (3) holds, but we do not know a priori this correspondence between each A_i and B_j. We seek to find X in this scenario.

The group of proper rigid-body motions, $SE(3)$, is a Lie group, and hence concepts of integration and convolution exist. If $H \in SE(3)$ is a generic 4×4 homogeneous transformation of the form in (1) where the rotation is parameterized in terms of Euler angles as $R = R_3(\alpha)R_1(\beta)R_3(\gamma)$ (where $R_i(\theta)$ is a counterclockwise rotation by θ around coordinate axis i) and the translation is $\mathbf{t} = [tx, ty, tz]^T$, then the 'natural' integral of any rapidly decaying function is computed as

$$\int_{SE(3)} f(H)\, dH = \int_{\mathbb{R}^3} \int_{SO(3)} f(H(R, \mathbf{t}))\, dR\, d\mathbf{t}$$

where $dR = \sin\beta d\alpha d\beta d\gamma$ and $d\mathbf{t} = dt_x dt_y dt_z$, with $-\infty < tx, ty, tz < \infty$ and $(\alpha, \beta, \gamma) \in [0, 2\pi] \times [0, \pi] \times [0, 2\pi]$. This integral is 'natural' in the sense that it is the unique one (up to scaling of the volume element by an arbitrary constant) such that

$$\int_{SE(3)} f(H)dH = \int_{SE(3)} f(H^{-1})dH = \int_{SE(3)} f(HH_0)dH = \int_{SE(3)} f(H_0H)dH \tag{4}$$

for any fixed $H_0 \in SE(3)$. This choice of integral, being invariant under shifts on the left and on the right in the above equation, is called the bi-invariant, or Haar, measure. The above instantiation of the bi-invariant integral for $SE(3)$ using Z-X-Z Euler angles and Cartesian coordinates for translation is not unique. Any parametrization of $SE(3)$ will do.

If $\int_{SE(3)} |f(H)|^p dH < \infty$ then we say $f \in L^p(SE(3))$. Most of our discussion will be limited to functions $f \in L^1(SE(3)) \cap L^2(SE(3))$, together with the special case of a Dirac delta function, which will be defined shortly. In this context, the convolution of two such functions is defined as

$$(f_1 * f_2)(H) = \int_{SE(3)} f_1(\mathcal{H}) f_2(\mathcal{H}^{-1}H) d\mathcal{H} \tag{5}$$

where $\mathcal{H} \in SE(3)$ is a dummy variable of integration.

A Dirac delta function can be defined for $SE(3)$ just like in the case of $\mathbb{R}^n$. It is defined by the properties $\int_{SE(3)} \delta(\mathcal{H})d\mathcal{H} = 1$ and $(f * \delta)(H) = f(\mathbb{I}_4)$ where $\mathbb{I}_4 = H(\mathbb{I}_3, \mathbf{0})$ is the 4×4 identity matrix (and the identity element of $SE(3)$), whereas $\mathbb{I}_3$ is the 3×3 identity. Intuitively a Dirac delta can be thought of as a function that has a spike with infinite height at the identity and vanishes everywhere else.

A shifted Dirac delta function can be defined as $\delta_A(H) = \delta(A^{-1}H)$ which places the spike at $A \in SE(3)$. The inverse operation, A^{-1}, must be applied to the argument of the function to move the spike from the identity to A.

2 Batch Solution

In this light we can think of (3) as the equation

$$(\delta_{A_i} * \delta_X)(H) = (\delta_X * \delta_{B_i})(H). \tag{6}$$

The addition of this mathematical terminology provides freedom to do something that we cannot do with (3). Namely, whereas the addition (as opposed to multiplication) of homogeneous transformation matrices is nonsensical, the addition of real-valued functions $f_1(H) + f_2(H)$ is a perfectly reasonable operation. And since convolution is a linear operation on functions, we can write all n instances of (6) into a single equation of the form

$$(f_A * \delta_X)(H) = (\delta_X * f_B)(H) \text{ where } f_A(H) = \frac{1}{n}\sum_{i=1}^{n} \delta(A_i^{-1}H) \tag{7}$$

and $f_B(H)$ is of a similar form computed from $\{B_j\}$.

When written in this way, it does not matter if we know the correspondence between each A_i and B_j. The above functions are normalized to be probability densities: $\int_{SE(3)} f_A(H)dH = \int_{SE(3)} f_B(H)dH = 1$. The functions $f_A(H)$ and $f_B(H)$ are not in $L^2(SE(3))$, but for our purposes this will not be a problem.

Let us assume that the set of A_i's and the set of B_j's are each clumped closely together. In other words, given a measure of distance between reference frames, $d : SE(3) \times SE(3) \to \mathbb{R}_{\geq 0}$, we have that $d(A_i, A_j), d(B_i, B_j) < \epsilon << 1$. This assumption can be made true for example, if we are using small relative motions between consecutive reference frames, regardless of whether the whole trajectory is long or not.

The convolution of "highly focused" distributions corresponding to closely clumped sets of reference frames have some interesting properties that we can exploit to solve for X. In particular, let the mean and covariance of a probability density $f(H)$ be defined by the conditions [8],[15]

$$\int_{SE(3)} \log(M^{-1}H)f(H)dH = \mathbb{O} \text{ and}$$

$$\Sigma = \int_{SE(3)} \log^{\vee}(M^{-1}H)[\log^{\vee}(M^{-1}H)]^T f(H)dH. \tag{8}$$

where explicit formulas for the matrix logarithm, $\log(H)$, and its vectorized form, $\log^{\vee}(H)$, are given in [8, 15]. The operation $\log(H)$ takes any element in $SE(3)$ (with rotational part that has an angle of rotation, θ, in the range $0 \leq \theta < \pi$) into the the corresponding unique element in the Lie algebra $se(3)$ such that $\exp(\log(H)) = H$, where $\exp(\cdot)$ is the matrix exponential. Since $SO(3)$ can be viewed as a solid three-dimensional ball of radius π with antipodal points identified, the exclusion of the 2D bounding sphere of radius π in $SO(3)$ defines a 5D set of measure zero in $SE(3)$ that has no effect on the computation of Σ in the above equation.

It is always the case that $\log(H) = \begin{pmatrix} \Omega & \mathbf{v} \\ \mathbf{0}^T & 0 \end{pmatrix}$ where $\Omega = -\Omega^T \in so(3)$. The map $\vee : se(3) \rightarrow \mathbb{R}^6$ is then composed with the $\log$ to give $\log^{\vee}(H) = [\boldsymbol{\omega}^T, \mathbf{v}^T]^T$ where $\boldsymbol{\omega} \in \mathbb{R}^3$ is the vector corresponding to Ω such that $\Omega\mathbf{x} = \boldsymbol{\omega} \times \mathbf{x}$ for any $\mathbf{x} \in \mathbb{R}^3$, where $\times$ is the vector cross product.

The definition of mean used above differs from that most often used in the literature when taking a Riemannian-geometric (rather than Lie-group) approach [11–14] which is of the form $M' = \arg\min_M \int_{SE(3)} [d(M,H)]^2 f(H) dH$ where $d(M,H)$ is a Riemannian distance function ($d(M,H) = \| \log(M^{-1}H) \|_W^2$, for example) and W is a weighting matrix related to the Riemannian metric tensor that is chosen. There are two reasons for our definition. First, in our definition there is no need to introduce a weighting matrix, and therefore we avoid coloring the result by an arbitrary choice. Second, in the context of robotics problems in which reference frames are attached to rigid links it is more natural in the following sense. If a single rigid link has a world frame attached to its base, and a reference frame attached to its distal end, and that distal reference frame is recorded at two different times as a joint at the base rotates, then the translation part of the average of these two reference frames should lie on the arc that joints the two. M will have this property, but M' will not. Having said this, if we were considering data on $SO(3)$ rather than $SE(3)$, and if $W = \mathbb{I}$ were chosen, the two definitions would become the same thing since for $SO(3)$ the distance (metric) function $d(R_1, R_2) \doteq \| \log(R_1^{-1} R) \|$ is bi-invariant and $Ad(R) = R$. But for $SE(3)$ neither of these statements are true: $Ad(H) \neq H$, and there does not exist a bi-invariant metric (though there does exist a bi-invariant integration measure).

If $f(H)$ is of the form of $f_A(H)$ given above, then discrete versions of (8) are

$$\sum_{i=1}^{n} \log(M_A^{-1} A_i) = \mathbb{O} \quad \text{and} \quad \Sigma_A = \frac{1}{n} \sum_{i=1}^{n} \log^{\vee}(M_A^{-1} A_i) [\log^{\vee}(M_A^{-1} A_i)]^T \tag{9}$$

where explicit formulas for $\log(H)$ and $\log^{\vee}(H)$ and $\exp$ are given in [8].

An iterative procedure for computing M_A was presented in [15] in which an initial estimate of the form $M_A^0 = \exp(\frac{1}{n} \sum_{i=1}^{n} \log(A_i))$ is chosen, and then a gradient descent procedure is used to update so as to minimize the cost $C(M) = \left\| \sum_{i=1}^{n} \log(M^{-1} A_i) \right\|^2$, and the minimum defines M_A.

It can be shown that if these quantities are computed for two highly focused functions, f_1 and f_2, that the same quantities for the convolution of these functions can be computed as [15]

$$M_{1*2} = M_1 M_2 \text{ and } \Sigma_{1*2} = Ad(M_2^{-1})\Sigma_1 Ad^T(M_2^{-1}) + \Sigma_2 \tag{10}$$

where $Ad(H) = \begin{pmatrix} R & \mathbb{O} \\ \hat{\mathbf{t}}R & R \end{pmatrix}$.

Here for any $\mathbf{a} \in \mathbb{R}^3$, $\hat{\mathbf{a}}$ is the skew-symmetric matrix such that $\hat{\mathbf{a}}\mathbf{b} = \mathbf{a} \times \mathbf{b}$. And by a slight abuse of notation, we use $\vee$ as the reverse map which gives $(\hat{\mathbf{a}})^\vee = \mathbf{a}$. The use of $\vee$ to denote maps from the Lie algebra $so(3)$ into $\mathbb{R}^3$ and from $se(3)$ into $\mathbb{R}^6$ should not be a source of confusion, as the version being used is defined by the argument to which it is applied.

The mean of $\delta_X(H)$ is $M_X = X$, and its covariance is the zero matrix. Therefore, (7) together with (10) gives two equations:

$$\text{(a)} \; \boxed{M_A X = X M_B} \;\text{ and }\; \text{(b)} \; \boxed{Ad(X^{-1})\Sigma_A Ad^T(X^{-1}) = \Sigma_B} \tag{11}$$

These two equations can be solved in a similar way to how the two equations $A_1 X = X B_1$ and $A_2 X = X B_2$ are solved.

First, we seek the rotational component, R_X, of X. From (11a) we have that,

$$\mathbf{n}_{M_A} = R_X \mathbf{n}_{M_B} \tag{12}$$

where $\mathbf{n}_H$ is the direction of the screw axis of the homogeneous transform H [1].

If we decompose Σ_{M_A} and Σ_{M_B} into blocks as $\Sigma_i = \begin{pmatrix} \Sigma_i^1 & \Sigma_i^2 \\ \Sigma_i^3 & \Sigma_i^4 \end{pmatrix}$ where $\Sigma_i^3 = (\Sigma_i^2)^T$, then we can take the first two blocks of (11b) and write

$$\Sigma_{M_B}^1 = R_X^T \Sigma_{M_A}^1 R_X \text{ and } \Sigma_{M_B}^2 = R_X^T \Sigma_{M_A}^1 R_X (\widehat{R_X^T t x}) + R_X^T \Sigma_{M_A}^2 R_X \tag{13}$$

We can then find the eigendecomposition, $\Sigma_i = Q_i \Lambda Q_i^T$, where Q_i is the square matrix whose ith column is the eigenvector of Σ_i and Λ is the diagonal matrix with corresponding eigenvalues as diagonal entries and write the first block equation of (13) as,

$$\Lambda = Q_{M_B}^T R_X^T Q_{M_A} \Lambda Q_{M_A}^T R_X Q_{M_B} = \mathcal{Q}\Lambda\mathcal{Q}^T \tag{14}$$

The set of $\mathcal{Q}$s that satisfy this equation is given as,

$$\mathcal{Q} = \left\{ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \right\}$$

with the simple condition that Q_i is constrained to be a rotation matrix. This means that the rotation component of X is given by,

$$R_x = Q_{M_A} \mathcal{Q} Q_{M_B}^T \tag{15}$$

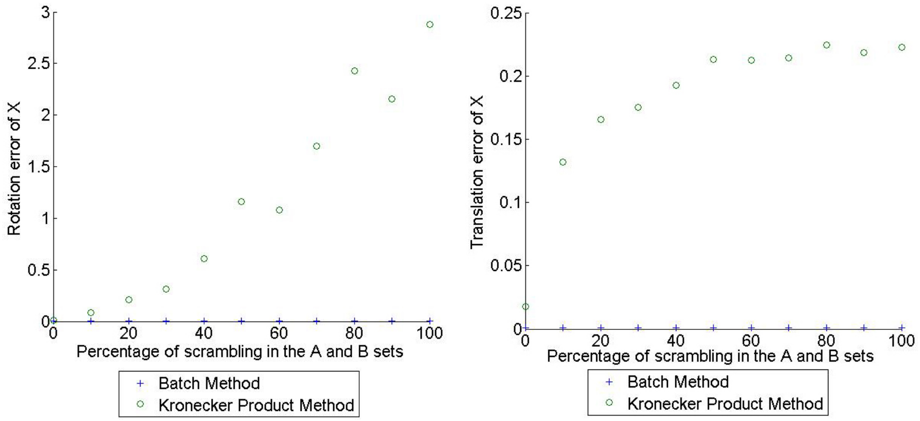

Fig. 1. Solution error with increasing permutations

The correct solution, from the set of 4 possibilities of R_X (15) can be found by applying (12) and choosing the one that minimizes $\|\mathbf{n}_{M_A} - R_X \mathbf{n}_{M_B}\|$. Once R_X is found, $\mathbf{t}_X$ can be found easily from blocks the 2 and 4 of (11)(b).

3 Applying the Batch Method

A traditional least-squares solution to the $AX = XB$ problem, for given multiple pairs (A_i, B_i), is to use a method based on the Kronecker product [16].

Using properties of the Kronecker product,$\otimes$, (3) can be written as

$$\begin{bmatrix} \mathbb{I}_9 - R_{B_i} \otimes R_{A_i} & \mathbb{O}_{9\times3} \\ \\ \mathbf{t}_{B_i}^T \otimes \mathbb{I}_3 & \mathbb{I}_3 - R_{A_i} \end{bmatrix} \begin{pmatrix} \text{vec}(R_X) \\ \\ \mathbf{t}_X \end{pmatrix} = \begin{pmatrix} \mathbf{0}_9 \\ \\ \mathbf{t}_{A_i} \end{pmatrix} \tag{16}$$

where $\text{vec}(R_X) \in \mathbb{R}^9$ is the vector obtained by stacking the columns of R_X, $\mathbb{I}_m$ is the $m \times m$ identity, $\mathbb{O}_{m\times n}$ is the $m \times n$ zero matrix, and $\mathbf{0}_n$ is the n-dimensional zero vector. By stacking multiple such equations for different pairs (A_i, B_i), a least-squares solution can then be found using SVD methods or using a pseudo-inverse. This solution can be easily projected back into the group using well known methods [16]. We use this common approach as a baseline to compare the performance of our method.

In the numerical experiments in this section, we fix a true baseline X (which is a priori unknown to our algorithm), generate a series of small motion B_i's, and for each compute $A_i = (XB_iX^{-1})$. We then scramble some fraction of the correspondences and compare our algorithm with the Kronecker solver.

The Batch method can handle any level of permutation (ie. non-correspondence) placed on the data sets of A and B. Figure 1 shows the error of the computed X for both the Kronecker Product solver and the Batch Method as a function of the amount of scrambling. The rotational and translational error are measured as $\| \log^\vee(R^T_{X_{calculated}} R_{X_{true}})\|$ and $\|\mathbf{t}_{X_{calculated}} - \mathbf{t}_{X_{true}}\|$ respectively.

It can easily be seen that while the Kronecker Product Solver solution degenerates quickly with even a slight permutation of the As and Bs, the Batch Solver finds the correct solution with any amount of permutation.

4 Conclusions

We establish that the $AX = XB$ sensor calibration problem can be solved with a new "Batch Method" that does not require a priori knowledge of the A and B correspondence. The Batch Method is shown to solve for X for any level of mismatch between As and Bs, performing much better than traditional methods, which require correspondences, under similar circumstances.

References

1. Chen, H.H.: A Screw Motion Approach to Uniqueness Analysis of Head-Eye Geometry. In: IEEE Conference on CVPR, pp. 145–151 (1991)
2. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-Squares Fitting of Two 3-D Point Sets. IEEE Trans. on Pattern Analysis and Machine Intel. 9(5), 698–700 (1987)
3. Park, F.C., Martin, B.J.: Robot Sensor Calibration: Solving $AX = XB$ on the Euclidean Group. IEEE Trans. Robotics and Automation 10(5), 717–721 (1994)
4. Shiu, Y.C., Ahmad, S.: Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form $AX = XB$. IEEE Trans. Robotics and Automation 5(1), 16–29 (1989)
5. Dong, W., Rostoucher, D., Gauthier, M.: Kinematics Parameters Estimation for an AFM/Robot Integrated Micro-Force Measurement System. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, October 18-22, pp. 6143–6148 (2010)
6. Kim, S.-J., Jeong, M.-H., Lee, J.-J., Lee, J.-Y., Kim, K.-G., You, B.-J., Oh, S.-R.: Robot Head-Eye Calibration Using the Minimum Variance Method. In: IEEE International Conference on Robotics and Biomimetics, December 14-18, pp. 1446–1451 (2010)
7. Dai, Y., Trumpf, J., Li, H., Barnes, N., Hartley, R.: Rotation Averaging with Application to Camera-Rig Calibration. In: Zha, H., Taniguchi, R.-I., Maybank, S. (eds.) ACCV 2009, Part II. LNCS, vol. 5995, pp. 335–346. Springer, Heidelberg (2010)
8. Chirikjian, G.S.: Stochastic Models, Information Theory, and Lie Groups, vol. 2. Birkhäuser, Basel (2011)
9. Mills, D.L.: Internet Time Synchronization: Network Time Protocol. IEEE Transactions on Communications 39(10), 1482–1493 (1991)
10. Mair, E., Fleps, M., Suppa, M., Burschka, D.: Spatio-temporal initialization for IMU to camera registration. In: 2011 IEEE ROBIO, pp. 557–564 (December 2011)
11. Pennec, X.: Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. Jour. of Mathematical Imaging and Vision 25(1), 127–154 (2006)

12. Tron, R., Vidal, R., Terzis, A.: Distributed pose averaging in camera networks via consensus on SE(3). In: Second ACM/IEEE ICDSC, pp. 1–10 (September 7, 2008)
13. Fletcher, P.T., Joshi, S., Lu, C., Pizer, S.M.: Gaussian Distributions on Lie Groups and Their Application to Statistical Shape Analysis. In: Taylor, C.J., Noble, J.A. (eds.) IPMI 2003. LNCS, vol. 2732, pp. 450–462. Springer, Heidelberg (2003)
14. Glover, J., Rus, D., Roy, N.: Probabilistic models of object geometry for grasp planning. In: Brock, O., Trinkle, J.C., Ramos, F. (eds.) Robotics Science and Systems IV. MIT Press (June 2009)
15. Wang, Y., Chirikjian, G.S.: Nonparametric Second-Order Theory of Error Propagation on the Euclidean Group. International Journal of Robotics Research 27(11-12), 1258–1273 (2008)
16. Boctor, E.M.: Enabling Technologies for Ultrasound Imagin. In: Computer-Assisted Intervention, CS Department, Johns Hopkins University. Thesis (2006)