Coregistration of Pose Measurement Devices Using Nonlinear Least Squares Parameter Estimation

Milan Ikits

UUCS-00-018

Virtual Reality Laboratory
Scientific Computing and Imaging Institute
School of Computing
University of Utah
50 Central Campus Drive Rm 3490
Salt Lake City, UT 84112-9205, USA
http://www.sci.utah.edu/research/vr

December 15, 2000

Abstract

Multimodal visual haptic user interfaces can be made more effective by accurately colocating the workspaces of their components. We have developed a coregistration technique for pose measurement devices based on nonlinear least squares parameter estimation. A reduced quaternion parameterization is used for representing the orientation component of coordinate transformations, which avoids the numerical instability of traditional approaches. The method is illustrated with two examples: the colocation of a haptic device with a position tracker, and the coregistration of an optical and a magnetic tracking system.

1 Introduction

The combined use of immersive visual and haptic displays has received focused attention in the last couple of years [12, 4, 15, 17, 2]. We believe that the precise colocation of the visual and haptic components is necessary to make these systems effective and useful for scientific exploration. For projection-based immersive displays, coregistration typically involves determining the following three coordinate transformations:

- The relative transformation between the coordinate bases of the position tracker and the haptic device.
- The position, orientation, and size of the display surface(s) with respect to a common base.
- The location of the user's eyes relative to the tracked head frame.

In this paper, we generalize the first problem and present a technique to coregister *pose measurement devices*, i.e. devices that measure position and orientation components of a moving coordinate frame relative to a fixed base.[1] The method can also be used to calibrate a tracking system from measurements taken with a more accurate device [7].

The coregistraton technique is based on attaching a rigid mechanical link between the two (or more) measured locations (Figure 1). We can determine the relative transformation between the devices by taking a series of simultaneous measurements and fitting a parametric model to the collected data. The parameters include the position and orientation of one base relative to the other and certain position and orientation components between the tracked frames. More precise parameter estimates can be obtained by considering inaccuracies in the measurements from both devices using a *total least squares* regression algorithm. It is also important to quantify how well the estimation algorithm performs taking into account the quality of the collected data.

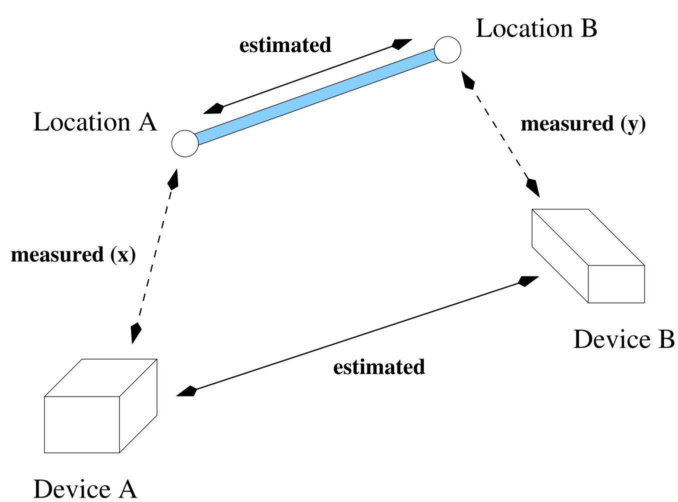

Figure 1. The idea behind the coregistration technique.

After briefly introducing the mathematical framework behind parameter estimation, we present a novel orientation representation based on a reduced quaternion parameterization. The method is illustrated in two typical situations: the colocation of a haptic device with a position tracker, and the coregistration of an optical and a magnetic tracking system.

The formulation presented here is similar to those used in the field of *robot calibration*, a taxonomy of which can be found in [6].

[1] The more commonly used term is *tracking device* or *position tracker*. For a recent survey on tracking technology, see [13].

2 The Coregistration Problem

The relationship between the two devices and the corresponding measured locations can be described as a coupled virtual mechanism (Figure 1). The geometric model of this mechanism is composed of the coordinate transformations between the two stationary bases and the two (or more) moving coordinate frames. Some of these transformations are fix and have to be estimated (denoted by the solid arrows in the figure). Other transformations are measured by the devices (indicated by the dashed arrows). We collect the parameters that describe the fix transformations into a parameter vector $\boldsymbol{\phi}$. The virtual mechanism is expressed mathematically as a nonlinear system of equations. We adopt the traditional formulation by considering device A as input, and device B as output of this system:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \boldsymbol{\phi}) \tag{1}$$

i.e. the output measurements $\mathbf{y}$ are a function of the input measurements $\mathbf{x}$ and the model parameters $\boldsymbol{\phi}$. The two examples in section 4 will demonstrate how to obtain these equations from the actual geometric arrangement. Our goal is to estimate $\boldsymbol{\phi}$ from a series of input-output measurement pairs ($\hat{\mathbf{x}}^i$, $\hat{\mathbf{y}}^i$). To achieve this, we minimize the error between the output measurements and the model according to the *least squares* principle:

$$S = \sum_{i=1}^{N} \mathbf{e}_y(\mathbf{f}(\mathbf{x}^i, \boldsymbol{\phi}), \hat{\mathbf{y}}^i)^T \, \mathbf{W}_i \, \mathbf{e}_y(\mathbf{f}(\mathbf{x}^i, \boldsymbol{\phi}), \hat{\mathbf{y}}^i) \tag{2}$$

where $\mathbf{e}_y$ is the output *error metric*, which is most of the time defined as the difference between the model output and the output measurement vector:

$$\mathbf{e}_y(\mathbf{f}(\mathbf{x}, \boldsymbol{\phi}), \hat{\mathbf{y}}) = \mathbf{f}(\mathbf{x}, \boldsymbol{\phi}) - \hat{\mathbf{y}} \tag{3}$$

and $\mathbf{W}_i$ is a weighting matrix, which we set to the inverse covariance of the output measurements:

$$\mathbf{W}_i = \mathbf{V}_i^{-1} \tag{4}$$

Since (2) is a nonlinear function of the parameters, this formulation belongs to the family of nonlinear ordinary least squares (OLS) problems [1].

A more sophisticated approach takes into account not only the output measurement errors, but also the input measurement errors, resulting in nonlinear weighted total least squares estimation (TLS) [16]:

$$S = \sum_{i=1}^{N} \begin{bmatrix} \mathbf{e}_x(\mathbf{x}^i, \hat{\mathbf{x}}^i) \\ \mathbf{e}_y(\mathbf{f}(\mathbf{x}^i, \boldsymbol{\phi}), \hat{\mathbf{y}}^i) \end{bmatrix}^T \mathbf{W}_i \begin{bmatrix} \mathbf{e}_x(\mathbf{x}^i, \hat{\mathbf{x}}^i) \\ \mathbf{e}_y(\mathbf{f}(\mathbf{x}^i, \boldsymbol{\phi}), \hat{\mathbf{y}}^i) \end{bmatrix} \tag{5}$$

Since all input measurements are considered as parameters of the system, the original parameters $\boldsymbol{\phi}$ are augmented resulting in a new parameter vector $\boldsymbol{\xi}$:

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \\ \boldsymbol{\phi} \end{bmatrix} \tag{6}$$

The implementation of both estimation techniques is discussed in detail in Appendix B.

3 Representation of Orientation

We need to represent the coordinate transformations with a minimum set of parameters to avoid having to use a constrained optimization algorithm. The position components are parameterized by their Cartesian coordinates expressed in the frame they originate from. For the orientation components, however, we face the problem of choosing an appropriate representation. Euler angles are minimal but suffer from singularities, otherwise known as gimbal locks. The exponential map is also minimal, but besides coping with singularities, it is difficult to use it for combining rotations [5]. Rotation matrices are highly redundant and it would be difficult to keep track of the large number of constraints they are subject to, or to come up with a reduced parameterization. Quaternions are an optimal choice, because they only require a single constraint. It is also possible to represent them in a reduced form, which is more convenient and elegant to include in the geometric model than Euler angles are. Since there is more than one way to represent a quaternion with three parameters only [8], we need to examine certain conditions at every iteration step of the estimation procedure and dynamically reparameterize the system to avoid singularities.

3.1 Reduced Quaternion Parameterization

The reduced representation is based on the observation that if we know the three smallest magnitude elements of a quaternion and the sign of the largest magnitude element, we can compute this element from the constraint equation. Let's denote the minimum magnitude vector component of quaternion $\boldsymbol{q}$ by $\mathbf{s}$. Without loss of generality we can assume that:

$$\mathbf{s} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{7}$$

i.e. the largest magnitude element of $\boldsymbol{q} = [\ q_0\ \ q_1\ \ q_2\ \ q_3\ ]^T$ is q_0. Since valid rotations are represented by unit quaternions, we calculate q_0 as:

$$q_0 = \pm\sqrt{1 - \mathbf{s}^T\mathbf{s}} = \pm\sqrt{1 - q_1^2 - q_2^2 - q_3^2} \tag{8}$$

Thus, we can define an operator that converts the reduced quaternion to the full representation:

$$\boldsymbol{q}(\mathbf{s}) = \begin{bmatrix} \pm\sqrt{1 - q_1^2 - q_2^2 - q_3^2} \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{9}$$

The Jacobian of this operator $\partial\boldsymbol{q}/\partial\mathbf{s}$ can be written in the following form:

$$\frac{\partial\boldsymbol{q}}{\partial\mathbf{s}} = \begin{bmatrix} -q_1/q_0 & -q_2/q_0 & -q_3/q_0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

This matrix is well-conditioned as long as q_0 is not close to zero, which is true, since q_0 is assumed to be the largest magnitude element of $\boldsymbol{q}$. The sign of q_0 is fixed until we choose a different reduced representation. Notice that there are four possible reduced forms, much fewer than the number of different Euler angles representations. The ambiguity in (9) can be handled by carrying the sign over from the previous iteration of the estimation procedure.

3.2 Error Metric

A related problem is how to calculate the difference between two orientations. Since the space of 3DOF rotations SO(3) is nonlinear, this is not as intuitive to define as position error. If the difference is small, linearized forms can be used. These include direct difference for Euler angles and quaternions, and differential orthogonal rotations for rotation matrices [6].

There is an alternative formula for quaternions, which closely approximates the ideal definition, i.e. the angle of rotation. It is based on the quaternion $\boldsymbol{q}_e$ that rotates between the two orientations $\boldsymbol{q}_A$ and $\boldsymbol{q}_B$:

$$\boldsymbol{q}_A \, \boldsymbol{q}_e = \boldsymbol{q}_B \tag{11}$$

from which:

$$\boldsymbol{q}_e = \boldsymbol{q}_A^* \, \boldsymbol{q}_B \tag{12}$$

where $*$ denotes the quaternion conjugate operator [9]. We can use the vector part of $\boldsymbol{q}_e$ as the error metric:

$$\mathbf{e}(\boldsymbol{q}_A, \boldsymbol{q}_B) = \mathbf{q}_e = q_A \, \mathbf{q}_B - q_B \, \mathbf{q}_A - \mathbf{q}_A \times \mathbf{q}_B \tag{13}$$

since it contains information about both the angle θ_e and the axis $\mathbf{k}_e$ of rotation:

$$\mathbf{q}_e = \mathbf{k}_e \sin(\frac{\theta_e}{2}) \approx \mathbf{k}_e \frac{\theta_e}{2} \tag{14}$$

The optimization procedure essentially minimizes the norm square of $\mathbf{q}_e$, which is a quantity very close to the square of the rotation half-angle, if the angle is small. We also need the partial derivative of $\mathbf{q}_e$ with respect to $\boldsymbol{q}_A$:

$$\frac{\partial \mathbf{q}_e}{\partial \boldsymbol{q}_A} = \begin{bmatrix} \mathbf{q}_B & \mathbf{S}(\mathbf{q}_B) - q_B \mathbf{I} \end{bmatrix} \tag{15}$$

where $\mathbf{S}(\mathbf{q})$ is a skew-symmetric matrix that represents the cross product by $\mathbf{q}$.

In practice, we found that it does not significantly matter if we use this definition or just the vector difference between the reduced quaternion parameters. We include it here for completeness and also formulate the estimation procedures with an error metric in mind.

4 Examples

To illustrate the use of the mathematical framework, two example formulations are presented. In the first, we coregister two full pose measurement devices, such as a haptic device and a 6DOF tracking system. The second example shows how to colocate a 6DOF full pose and a 3DOF position only tracker using multiple measurement locations.

4.1 Full Pose Coregistration

In this case the devices measure all six degrees of freedom of the tracked frames relative to their bases. We consider device A the input, and device B the output of the system (Figure 2). The transformation between the base frames A_0 and B_0 is represented by displacement vector $\mathbf{p}_0$ and quaternion $\boldsymbol{q}_0$. The relative position and orientation of the tracked frames A_1 and B_1 are expressed by $\mathbf{p}_1$ and $\boldsymbol{q}_1$. Let's find the position and orientation of frame B_1 with respect to frame B_0 via the B_0-A_0-A_1-B_1 loop:

$$\mathbf{y} = \mathbf{f}(\boldsymbol{\phi}, \mathbf{x}) = \begin{bmatrix} \mathbf{p}_0 + \mathbf{R}(\boldsymbol{q}(\mathbf{s}_0))(\mathbf{p}_A + \mathbf{R}(\boldsymbol{q}(\mathbf{s}_A))\mathbf{p}_1) \\ \boldsymbol{q}(\mathbf{s}_0) \, \boldsymbol{q}(\mathbf{s}_A) \, \boldsymbol{q}(\mathbf{s}_1) \end{bmatrix} \tag{16}$$

from which:

$$\boldsymbol{\phi} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{s}_0 \\ \mathbf{p}_1 \\ \mathbf{s}_1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{p}_A \\ \mathbf{s}_A \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{p}_B \\ \boldsymbol{q}_B \end{bmatrix} \tag{17}$$

and $\mathbf{R}(\boldsymbol{q})$ is an operator that converts a quaternion to a rotation matrix (Appendix A). Alternatively, we could omit the orientation part in (16) and obtain a partial formulation:

$$\mathbf{y} = \mathbf{f}(\boldsymbol{\phi}, \mathbf{x}) = \left[\; \mathbf{p}_0 + \mathbf{R}(\boldsymbol{q}(\mathbf{s}_0))(\mathbf{p}_A + \mathbf{R}(\boldsymbol{q}(\mathbf{s}_A))\mathbf{p}_1) \; \right] \tag{18}$$

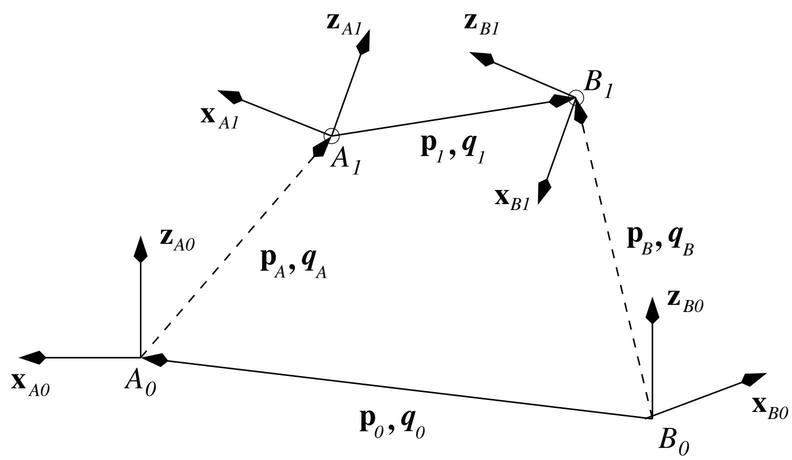

Figure 2. Coregistration of two full pose measurement devices.

4.2 Partial Pose Coregistration

This example shows how to formulate the coregistration equations for a 6DOF full pose and a 3DOF position tracker (Figure 3). Four measurement locations are used with the purpose of being able to extract full pose information from three simultaneous position measurements [7]. Thus, three attachments are needed, which are joined at location A_1. The equations are obtained by finding the position of B_j with respect to frame B_0 via the B_0-A_0-A_1-B_j loop (j = 1, 2, 3):

$$\mathbf{y} = \mathbf{f}(\boldsymbol{\phi}, \mathbf{x}) = \begin{bmatrix} \mathbf{p}_0 + \mathbf{R}(\boldsymbol{q}(\mathbf{s}_0))(\mathbf{p}_A + \mathbf{R}(\boldsymbol{q}(\mathbf{s}_A))\mathbf{p}_1) \\ \mathbf{p}_0 + \mathbf{R}(\boldsymbol{q}(\mathbf{s}_0))(\mathbf{p}_A + \mathbf{R}(\boldsymbol{q}(\mathbf{s}_A))\mathbf{p}_2) \\ \mathbf{p}_0 + \mathbf{R}(\boldsymbol{q}(\mathbf{s}_0))(\mathbf{p}_A + \mathbf{R}(\boldsymbol{q}(\mathbf{s}_A))\mathbf{p}_3) \end{bmatrix} \tag{19}$$

from which:

$$\boldsymbol{\phi} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{s}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{p}_A \\ \mathbf{s}_A \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{p}_{B1} \\ \mathbf{p}_{B2} \\ \mathbf{p}_{B3} \end{bmatrix} \tag{20}$$

5 Conclusions

To verify the feasibility of the technique, we have implemented and tested the two examples above in GNU Octave [3]. The second example was used to coregister a magnetic and an optical tracking system in a real experiment [7]. In practice, both estimation algorithms are needed: OLS is used to get a "rough" estimate, which can be refined later with TLS. Note, however, that if the input noise is not significant relative to the output noise, both OLS and TLS will yield very similar estimates.

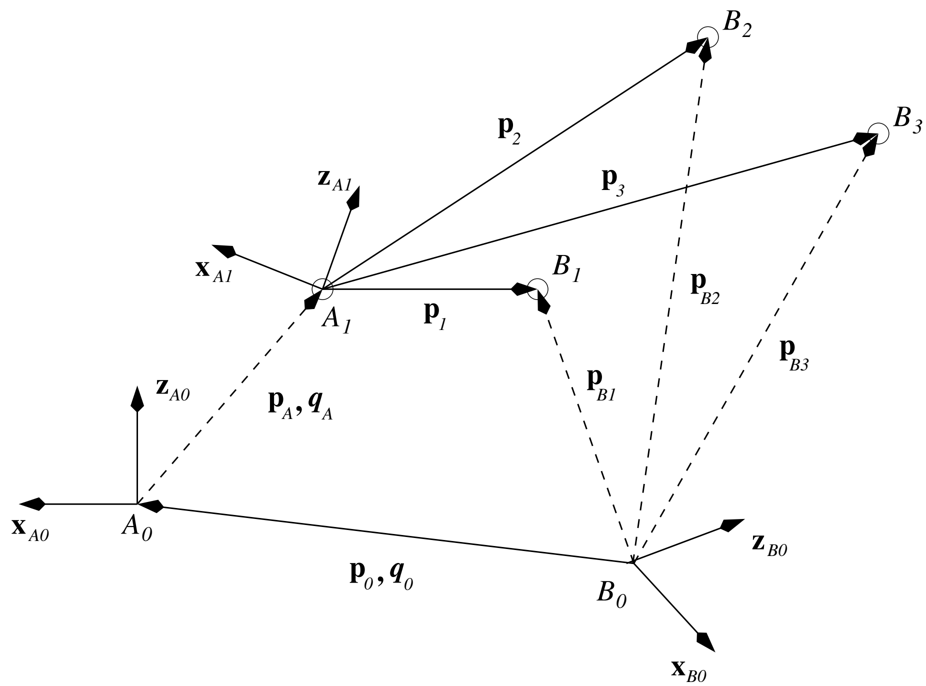

Figure 3. Colocation of a 6DOF full pose and a 3DOF position tracker.

A Operator $\mathbf{R}(\boldsymbol{q})$

The following operator transforms a quaternion to the corresponding rotation matrix [9]:

$$\mathbf{R}(\boldsymbol{q}) = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2q_1q_2-2q_0q_3 & 2q_1q_3+2q_0q_2 \\ 2q_1q_2+2q_0q_3 & q_0^2+q_2^2-q_1^2-q_3^2 & 2q_2q_3-2q_0q_1 \\ 2q_1q_3-2q_0q_2 & 2q_2q_3+2q_0q_1 & q_0^2+q_3^2-q_1^2-q_2^2 \end{bmatrix} \tag{21}$$

Note that this operator has a more commonly used equivalent form, which is obtained by incorporating the constraint $q_0^2+q_1^2+q_2^2+q_3^2=1$. In the estimation procedures the partial derivatives are frequently used. These can be derived directly from (21):

$$\frac{\partial \mathbf{R}}{\partial q_0} = 2\begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \tag{22}$$

$$\frac{\partial \mathbf{R}}{\partial q_1} = 2\begin{bmatrix} q_1 & q_2 & q_3 \\ q_2 & -q_1 & -q_0 \\ q_3 & q_0 & -q_1 \end{bmatrix} \tag{23}$$

$$\frac{\partial \mathbf{R}}{\partial q_2} = 2\begin{bmatrix} -q_2 & q_1 & q_0 \\ q_1 & q_2 & q_3 \\ -q_0 & q_3 & -q_2 \end{bmatrix} \tag{24}$$

$$\frac{\partial \mathbf{R}}{\partial q_3} = 2\begin{bmatrix} -q_3 & -q_0 & q_1 \\ q_0 & -q_3 & q_2 \\ q_1 & q_2 & q_3 \end{bmatrix} \tag{25}$$

B Nonlinear Least Squares Parameter Estimation

The problems we are interested in can be modeled by the following nonlinear vector equation:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \boldsymbol{\phi}) \tag{26}$$

where $(\mathbf{x}, \mathbf{y})$ stand for the input and output variables of the problem, and $\boldsymbol{\phi}$ is the parameter vector.

In order to get an estimate of $\boldsymbol{\phi}$, we take N measurements and use an iterative algorithm based on the principle of *least squares*. Let $(\mathbf{x}_*^i, \mathbf{y}_*^i)$ denote the actual but unknown input and output variables at measurement i, and $\boldsymbol{\phi}_*$ the actual but unknown parameter vector. Then, assuming a perfect model, it is true that:

$$\mathbf{y}_*^i = \mathbf{f}(\mathbf{x}_*^i, \boldsymbol{\phi}_*), \;\; i = 1, \ldots, N \tag{27}$$

In reality, the measurements are corrupted by noise, which we express through error metrics $\mathbf{e}_x(\mathbf{x}^i, \hat{\mathbf{x}}^i)$ and $\mathbf{e}_y(\mathbf{y}^i, \hat{\mathbf{y}}^i)$. For our purposes, these functions consist of the vector difference of the position components and the metric introduced in section 3.2 for the orientation measurements.

B.1 Weighted Ordinary Least Squares

Let the output measurement errors $\mathbf{e}_y(\mathbf{y}_*^i, \hat{\mathbf{y}}^i)$ be independent and normally distributed with zero mean and covariance matrices $\mathbf{V}_i$. The estimate of $\boldsymbol{\phi}_*$ is chosen to minimize the following expression:

$$S = \sum_{i=1}^{N} \mathbf{e}_y(\mathbf{f}(\mathbf{x}^i, \boldsymbol{\phi}), \hat{\mathbf{y}}^i)^T \, \mathbf{W}_i \, \mathbf{e}_y(\mathbf{f}(\mathbf{x}^i, \boldsymbol{\phi}), \hat{\mathbf{y}}^i) \tag{28}$$

where $\mathbf{W}_i$ is a positive definite, symmetric weighting matrix, which we set to the inverse covariance of the output measurement errors:

$$\mathbf{W}_i = \mathbf{V}_i^{-1} = \mathbf{R}_i^T \mathbf{R}_i \tag{29}$$

The upper triangular matrix $\mathbf{R}_i$ is obtained from the the Cholesky factorization of $\mathbf{W}_i$ [11]. We reformulate the minimization problem by collecting the weighted measurement errors into a single vector $\mathbf{g}(\boldsymbol{\phi})$:

$$\mathbf{g}(\boldsymbol{\phi}) = \begin{bmatrix} \mathbf{g}_1(\boldsymbol{\phi}) \\ \vdots \\ \mathbf{g}_N(\boldsymbol{\phi}) \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \; \mathbf{e}_y(\mathbf{f}(\mathbf{x}^1, \boldsymbol{\phi}), \hat{\mathbf{y}}^1) \\ \vdots \\ \mathbf{R}_N \; \mathbf{e}_y(\mathbf{f}(\mathbf{x}^N, \boldsymbol{\phi}), \hat{\mathbf{y}}^N) \end{bmatrix} \tag{30}$$

Hence, the objective function (28) is expressed as:

$$S = \mathbf{g}(\boldsymbol{\phi})^T \mathbf{g}(\boldsymbol{\phi}) \tag{31}$$

To find the optimal parameter vector $\boldsymbol{\phi}$, we use an iterative procedure, which linearizes $\mathbf{g}$ around the current estimate $\boldsymbol{\phi}_k$ of step k:

$$S = (\mathbf{g}(\boldsymbol{\phi}_k) + \mathbf{J}\boldsymbol{\Delta\phi})^T (\mathbf{g}(\boldsymbol{\phi}_k) + \mathbf{J}\boldsymbol{\Delta\phi}) \tag{32}$$

The parameter Jacobian of $\mathbf{g}$ is expressed as:

$$\mathbf{J} = \frac{\partial \mathbf{g}(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \mathbf{R}_1 \; \mathbf{E}_y(\mathbf{f}(\mathbf{x}^1, \boldsymbol{\phi}_k), \hat{\mathbf{y}}^1) \; \mathbf{F}(\mathbf{x}^1, \boldsymbol{\phi}_k) \\ \vdots \\ \mathbf{R}_N \; \mathbf{E}_y(\mathbf{f}(\mathbf{x}^N, \boldsymbol{\phi}_k), \hat{\mathbf{y}}^N) \; \mathbf{F}(\mathbf{x}^N, \boldsymbol{\phi}_k) \end{bmatrix} \tag{33}$$

where:

$$\mathbf{E}_y = \frac{\partial \mathbf{e}_y}{\partial \mathbf{y}} \tag{34}$$

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\phi}} \tag{35}$$

The solution to minimizing (32) is the well-known formula for linear least squares problems [11]:

$$\boldsymbol{\Delta\phi} = -(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\mathbf{g}(\phi_k) \tag{36}$$

from which $\boldsymbol{\phi}_k$ is updated:

$$\boldsymbol{\phi}_{k+1} = \boldsymbol{\phi}_k + \boldsymbol{\Delta\phi} \tag{37}$$

and the procedure is repeated until the corrections are sufficiently small:

$$\| \boldsymbol{\Delta\phi} \| < \varepsilon \tag{38}$$

B.2 Weighted Total Least Squares

So far we have neglected the input measurement errors, which is not usually a problem as long as they are small relative to the output measurement errors. It can be shown, however, that the parameter estimates might be biased if the input measurement errors are sufficiently large and are not included in the estimation procedure [10]. To avoid this problem, more complicated formulations that treat the input and output measurements equally have been developed [14, 16, 18].

Some of these methods are based on treating the input measurements as parameters of the problem. Let the measurement errors $\mathbf{e}_x(\hat{\mathbf{x}}^i - \mathbf{x}_*^i)$, $\mathbf{e}_y(\hat{\mathbf{y}}_i - \mathbf{y}_*^i)$ be independent and normally distributed with zero mean and joint covariance matrices $\mathbf{V}_i$. Then the estimates $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N, \boldsymbol{\phi}$ are chosen to minimize the following expression:

$$S = \sum_{i=1}^{N} \begin{bmatrix} \mathbf{e}_x(\mathbf{x}^i, \hat{\mathbf{x}}^i) \\ \mathbf{e}_y(\mathbf{f}(\mathbf{x}^i, \boldsymbol{\phi}), \hat{\mathbf{y}}^i) \end{bmatrix}^T \mathbf{W}_i \begin{bmatrix} \mathbf{e}_x(\mathbf{x}^i, \hat{\mathbf{x}}^i) \\ \mathbf{e}_y(\mathbf{f}(\mathbf{x}^i, \boldsymbol{\phi}), \hat{\mathbf{y}}^i) \end{bmatrix} \tag{39}$$

We assume that the errors between measurements are independent, but those of a single measurement are not necessarily so. Thus, $\mathbf{W}_i$ is a positive definite, symmetric weighting matrix:

$$\mathbf{W}_i = \mathbf{V}_i^{-1} = \mathbf{R}_i^T \mathbf{R}_i \tag{40}$$

As before, $\mathbf{R}_i$ is obtained from the the Cholesky factorization of $\mathbf{W}_i$ and has the following form:

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{R}_{1,i} & \mathbf{R}_{2,i} \\ \mathbf{0} & \mathbf{R}_{3,i} \end{bmatrix} \tag{41}$$

We recast the minimization problem by collecting the input measurement and parameter estimates into a single vector $\boldsymbol{\xi}$:

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^1 \\ \vdots \\ \mathbf{x}^N \\ \boldsymbol{\phi} \end{bmatrix} \tag{42}$$

Hence, our goal is to minimize:

$$S = \mathbf{g}(\boldsymbol{\xi})^T \mathbf{g}(\boldsymbol{\xi}) \tag{43}$$

where:

$$\mathbf{g}(\boldsymbol{\xi}) = \begin{bmatrix} \mathbf{R}_{1,1}\ \mathbf{e}_x(\mathbf{x}^1, \hat{\mathbf{x}}^1) + \mathbf{R}_{2,1}\ \mathbf{e}_y(\mathbf{f}(\mathbf{x}^1, \boldsymbol{\phi}), \hat{\mathbf{y}}^1) \\ \vdots \\ \mathbf{R}_{1,N}\ \mathbf{e}_x(\mathbf{x}^N, \hat{\mathbf{x}}^N) + \mathbf{R}_{2,N}\ \mathbf{e}_y(\mathbf{f}(\mathbf{x}^N, \boldsymbol{\phi}), \hat{\mathbf{y}}^N) \\ \mathbf{R}_{3,1}\ \mathbf{e}_y(\mathbf{f}(\mathbf{x}^1, \boldsymbol{\phi}), \hat{\mathbf{y}}^1) \\ \vdots \\ \mathbf{R}_{3,N}\ \mathbf{e}_y(\mathbf{f}(\mathbf{x}^N, \boldsymbol{\phi}), \hat{\mathbf{y}}^N) \end{bmatrix} \tag{44}$$

The iterative procedure at each step calculates the correction $\boldsymbol{\Delta\xi}$, which is added to the current value of $\boldsymbol{\xi}$:

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \boldsymbol{\Delta\xi} \tag{45}$$

by finding the minimum of the linearized expression:

$$S = (\mathbf{g}(\boldsymbol{\xi}_k) + \mathbf{J}\boldsymbol{\Delta\xi})^T (\mathbf{g}(\boldsymbol{\xi}_k) + \mathbf{J}\boldsymbol{\Delta\xi}) \tag{46}$$

The parameter Jacobian $\mathbf{J}$ of $\mathbf{g}$ is written as:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_u(\boldsymbol{\xi}) & \mathbf{J}_\phi(\boldsymbol{\xi}) \end{bmatrix} \tag{47}$$

where:

$$\mathbf{J}_u = \frac{\partial \mathbf{g}(\boldsymbol{\xi})}{\partial \mathbf{u}} = \begin{bmatrix} \mathbf{R}_{1,1}\ \mathbf{E}_x(\mathbf{x}_k^1, \hat{\mathbf{x}}^1) + \mathbf{R}_{2,1}\ \mathbf{E}_y(\mathbf{f}(\mathbf{x}_k^1, \boldsymbol{\phi}_k), \hat{\mathbf{y}}^1)\ \mathbf{F}_x(\mathbf{x}_k^1, \boldsymbol{\phi}_k) \\ \vdots \\ \mathbf{R}_{1,N}\ \mathbf{E}_x(\mathbf{x}_k^N, \hat{\mathbf{x}}^N) + \mathbf{R}_{2,N}\ \mathbf{E}_y(\mathbf{f}(\mathbf{x}_k^N, \boldsymbol{\phi}_k), \hat{\mathbf{y}}^1)\ \mathbf{F}_x(\mathbf{x}_k^1, \boldsymbol{\phi}_k) \\ \mathbf{R}_{3,1}\ \mathbf{E}_y(\mathbf{f}(\mathbf{x}_k^1, \boldsymbol{\phi}_k), \hat{\mathbf{y}}^1)\ \mathbf{F}_x(\mathbf{x}_k^1, \boldsymbol{\phi}_k) \\ \vdots \\ \mathbf{R}_{3,N}\ \mathbf{E}_y(\mathbf{f}(\mathbf{x}_k^N, \boldsymbol{\phi}_k), \hat{\mathbf{y}}^1)\ \mathbf{F}_x(\mathbf{x}_k^N, \boldsymbol{\phi}_k) \end{bmatrix} \tag{48}$$

$$\mathbf{J}_\phi = \frac{\partial \mathbf{g}(\boldsymbol{\xi})}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \mathbf{R}_{2,1}\ \mathbf{E}_y(\mathbf{f}(\mathbf{x}_k^1, \boldsymbol{\phi}_k), \hat{\mathbf{y}}^1)\ \mathbf{F}_\phi(\mathbf{x}_k^1, \boldsymbol{\phi}_k) \\ \vdots \\ \mathbf{R}_{2,N}\ \mathbf{E}_y(\mathbf{f}(\mathbf{x}_k^N, \boldsymbol{\phi}_k), \hat{\mathbf{y}}^1)\ \mathbf{F}_\phi(\mathbf{x}_k^1, \boldsymbol{\phi}_k) \\ \mathbf{R}_{3,1}\ \mathbf{E}_y(\mathbf{f}(\mathbf{x}_k^1, \boldsymbol{\phi}_k), \hat{\mathbf{y}}^1)\ \mathbf{F}_\phi(\mathbf{x}_k^1, \boldsymbol{\phi}_k) \\ \vdots \\ \mathbf{R}_{3,N}\ \mathbf{E}_y(\mathbf{f}(\mathbf{x}_k^N, \boldsymbol{\phi}_k), \hat{\mathbf{y}}^1)\ \mathbf{F}_\phi(\mathbf{x}_k^N, \boldsymbol{\phi}_k) \end{bmatrix} \tag{49}$$

and:

$$\mathbf{E}_x = \frac{\partial \mathbf{e}_x}{\partial \mathbf{x}} \tag{50}$$

$$\mathbf{E}_y = \frac{\partial \mathbf{e}_y}{\partial \mathbf{y}} \tag{51}$$

$$\mathbf{F}_x = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \tag{52}$$

$$\mathbf{F}_\phi = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\phi}} \tag{53}$$

We could proceed with the linear least squares solution like we did in the previous section, which works well for a small number of measurements. Notice, however, that by incorporating the input measurements into the parameter vector, we increased the size of the search space significantly. More efficient procedures exploit the sparse structure of the Jacobian $\mathbf{J}$ [14].

B.3 Incorporating a priori Parameter Estimates

If we have a priori knowledge about the parameter estimates, it is useful to include it in the estimation procedure. This is done by adding a term to the objective functions (28) and (39):

$$S_\phi = S + \mathbf{e}_\phi(\boldsymbol{\phi}, \hat{\boldsymbol{\phi}})^T \, \mathbf{W}_\phi \, \mathbf{e}_\phi(\boldsymbol{\phi}, \hat{\boldsymbol{\phi}}) \tag{54}$$

where $\hat{\boldsymbol{\phi}}$ is our a priori estimate of $\boldsymbol{\phi}^*$, such that $\mathbf{e}_\phi(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}^*)$ is normally distributed with zero mean and covariance matrix $\mathbf{V}_\phi$. The iterative procedure is carried out as before, except that now $\mathbf{g}$ and $\mathbf{J}$ are augmented by:

$$\mathbf{g}_\phi(\boldsymbol{\phi}) = \mathbf{R}_\phi \, \mathbf{e}_\phi(\boldsymbol{\phi}, \hat{\boldsymbol{\phi}}) \tag{55}$$

$$\mathbf{J}_\phi = \mathbf{R}_\phi \, \frac{\partial \mathbf{e}_\phi}{\partial \boldsymbol{\phi}} \tag{56}$$

B.4 Evaluating the Parameter Estimation

An important step of parameter estimation is to check that the statistical assumptions about the model are appropriate, and to find out how accurate the parameter estimates are.

B.4.1 Goodness of Fit

It can be shown that if the Gaussian assumptions are valid, the objective function S follows the χ^2 distribution with $\nu = M - P$ degrees of freedom, where M is the length of $\mathbf{g}$ and P is the number of parameters [11]. A sample S^* from this distribution is obtained by computing S with the converged values of $\boldsymbol{\phi}$:

$$S^* = \mathbf{g}(\boldsymbol{\phi}_k)^T \mathbf{g}(\boldsymbol{\phi}_k) \tag{57}$$

By comparing S^* to the χ^2 distribution, we can find the probability of getting this value in light of the assumptions. If this probability is very low, we conclude that the assumptions are invalid and need to be modified, typically via changing the measurement error covariance matrices [18].

B.4.2 Estimating the a posteriori Parameter Covariance

An estimate of the a posteriori parameter covariance matrix can be calculated from the Jacobian using the converged values of the parameters [10]:

$$\mathbf{V}_\phi = (\mathbf{J}(\boldsymbol{\phi}_k)^T \mathbf{J}(\boldsymbol{\phi}_k))^{-1} \tag{58}$$

By taking the square root of the diagonal entries of this matrix, we get an estimate for the standard deviation of the parameter errors.

Acknowledgments

The author would like to thank Gordon Kindlmann and Dean Brederson for the energetic and enlightening discussions on quaternions. Support for this research was kindly provided by NSF Grant ACI-9978063 and the DOE Advanced Visualization Technology Center (AVTC).

References

[1] Y. Bard. *Nonlinear Parameter Estimation*. Academic Press, London, 1974.

[2] J. D. Brederson, M. Ikits, C. Johnson, and C. Hansen. The visual haptic workbench. In *Proc. Fifth PHANToM Users Group Workshop*, Aspen, CO, Oct. 2000.

[3] GNU Octave. http://www.octave.org.

[4] B. Grant, A. Helser, and R. Taylor II. Adding force display to a stereoscopic head-tracked projection display. In *Proc. VRAIS '98*, pages 81–88, Atlanta, GA, Mar. 1998.

[5] F. Grassia. Practical parametrization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3):29–48, 1998.

[6] J. Hollerbach and C. Wampler. The calibration index and taxonomy of robot kinematic calibration methods. *International Journal of Robotics Research*, 15(6):573–591, 1996.

[7] M. Ikits, J. Brederson, C. Hansen, and J. Hollerbach. An improved calibration framework for electromagnetic tracking devices. In *Proc. IEEE Virtual Reality*, pages 63–70, Yokohama, Japan, Mar. 2001.

[8] A. Katz. Special rotation vectors – a means for transmitting quaternion information in three components. *Journal of Aircraft*, 30(1):148–150, 1993.

[9] J. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, Princeton, NJ, 1999.

[10] J. Norton. *An Introduction to Identification*. Academic Press, London, 1986.

[11] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.

[12] Reachin Technologies AB. http://www.reachin.se.

[13] J. Rolland, L. Davis, and Y. Baillot. *Augmented Reality and Wearable Computers*, chapter A Survey of Tracking Technology for Virtual Environments. Lawrence Erlbaum Press, Mahwah, NJ, 2000.

[14] H. Schwetlick and V. Tiller. Numerical methods for estimating parameters in nonlinear models with errors in the variables. *Technometrics*, 27(1):17–24, Feb. 1985.

[15] D. Stevenson, K. Smith, J. Mclaughlin, C. Gunn, J. Veldkamp, and M. Dixon. Haptic workbench: A multisensory virtual environment. In *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems VI*, volume 3639, pages 356–366, 1999. San Jose, CA.

[16] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*, volume 9 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1991.

[17] T. von Wiegand, D. Schloerb, and W. Sachtler. Virtual workbench: Near-field virtual environment system with applications. *Presence: Teleoperators and Virtual Environments*, 8(5):492–519, Oct. 1999.

[18] C. Wampler, J. Hollerbach, and T. Arai. An implicit loop method for kinematic calibration and its application to closed-chain mechanisms. *IEEE Transactions on Robotics and Automation*, 11(5):710–724, Oct. 1995.