

# A General Framework for Temporal Calibration of Multiple Proprioceptive and Exteroceptive Sensors

Jonathan Kelly and Gaurav S. Sukhatme

**Abstract** Fusion of data from multiple sensors can enable robust navigation in varied environments. However, for optimal performance, the sensors must be calibrated relative to one another. Full sensor-to-sensor calibration is a *spatiotemporal* problem: we require an accurate estimate of the relative timing of measurements for each pair of sensors, in addition to the 6-DOF sensor-to-sensor transform. In this paper, we examine the problem of determining the time delays between multiple proprioceptive and exteroceptive sensor data streams. The primary difficulty is that the correspondences between measurements from different sensors are unknown, and hence the delays cannot be computed directly. We instead formulate temporal calibration as a registration task. Our algorithm operates by aligning curves in a three-dimensional orientation space, and, as such, can be considered as a variant of Iterative Closest Point (ICP). We present results from simulation studies and from experiments with a PR2 robot, which demonstrate accurate calibration of the time delays between measurements from multiple, heterogeneous sensors.

## 1 Introduction

In modern navigation systems, sensors with complimentary modalities and error characteristics are frequently deployed together to improve navigation accuracy. Fusing data from multiple sensors offers several potential advantages, such as enhanced precision and robustness. However, to properly combine measurements in a single navigation frame, the individual sensors must be calibrated relative to one another.

---

Department of Computer Science  
University of Southern California  
Los Angeles, California, USA 90089-0681  
e-mail: {jonathsk,gaurav}@usc.edu  
<http://robotics.usc.edu/res1/>

Recent work has primarily addressed sensor-to-sensor *spatial* calibration [1–3], i.e., the task of determining the six degrees-of-freedom (6-DOF) rigid body transform between sensor reference frames. However, full sensor-to-sensor calibration is a *spatiotemporal* problem: accurate estimates of the relative timing of measurements from the sensors are also required. In multi-sensor systems, unknown delays between data streams may be introduced by differences in transmission time, signal preprocessing, integration and filtering, etc. If the delay values are incorrectly estimated (perhaps simply assumed to be zero), then any fusion algorithm will produce sub-optimal results. At best, this will degrade performance — in the worst case (e.g., when measurements from one sensor are lost for a period of time), significant navigation errors may result.

In this paper, we examine the problem of determining the relative time delays between measurements from multiple proprioceptive and exteroceptive sensors.<sup>1</sup> Examples of proprioceptive sensors include inertial measurement units (IMUs), joint encoders and odometers; examples of exteroceptive sensors include cameras and laser range finders. In some cases, it may be possible to directly synchronize the sensor outputs using a hardware timing signal. Our interest here is in low-cost, “black box” devices, where direct synchronization is not possible. We assume that the individual sensor clocks are accurate, but that there are constant but unknown time shifts (delays) between the measurements, when they arrive and are timestamped at a centralized receiver.

The classic problem of time delay estimation for, e.g., sonar signal processing, has been treated as one of finding the delay value which maximizes the cross-correlation between two signals [4]. In the framework of parameter estimation, it is possible to determine the time delay by estimating the coefficients of a suitable FIR filter [5], where the signal and its delayed version are assumed to have originally been generated by the same sensor. Our problem is different in several ways: first, many proprioceptive sensors provide *rate* information, which cannot be immediately compared with the position or orientation measurements from an exteroceptive sensor. Second, there is also, in general, an unknown rigid body transform between between each pair of sensors. Third, the signals often have different temporal resolutions — cameras, for example, usually capture images at 30 frames per second or less, while IMU data is often available at 100 Hz or more.

These characteristics have led us to develop an alternative approach, in which we consider time delay estimation as a *registration* problem. Our algorithm can be regarded as a variant of iterative closest point (ICP) [6, 7]. ICP is normally used to register spatial data from two or more range scans. The fundamental difficulty in registration is that the *correspondences* between points in consecutive scans are unknown. Given a coarse initial guess for the transform between the scans, ICP selects nearest-neighbor correspondences, using a suitable distance metric. In a following optimization step, the transform parameters which minimize sum of squared distances between the points are determined. This transform is then applied, and the

---

<sup>1</sup> We use the terms ‘temporal calibration’ and ‘time delay estimation’ synonymously throughout the paper.

next cycle begins with a new set of correspondences. The steps are iterated until convergence, bringing the scans into alignment.

Although typically applied to range or point cloud data, ICP can be adapted for our problem. In our case the data have varied temporal resolutions. We note that if two sensors are rigidly attached to the same moving platform, then both must undergo the same orientation change over any period of time. For an exteroceptive sensor, the change in orientation is measured with respect to an external reference (e.g., several visual landmarks in the case of a camera). For proprioceptive rate sensors (such as IMUs), the rate data must be integrated to determine the orientation change. After carrying out the necessary preprocessing, we obtain two sets of orientation measurements for each pair of sensors; these measurements trace out curves in a three-dimensional orientation space.<sup>2</sup> Every point on each curve has a corresponding timestamp, defining the time at which the measurement arrived at the receiver. By registering the orientation curves, we are able to use the timestamp values to estimate the relative delay between the data streams.

Our algorithm, which we call Time Delay Iterative Closest Point (TD-ICP), uses a total least squares cost function for registration. This allows us to incorporate in a principled way the uncertainty in the individual sensor measurements. We use orientation information only, instead of position information, to avoid any potential ambiguities that may arise when dealing with, e.g., gravity. In addition to an estimate of the time delay, TD-ICP produces an improved estimate of the relative orientation of the sensors — this information can then be used to bootstrap a spatial calibration algorithm.

The remainder of the paper is organized as follows. We discuss related work in Section 2. In Section 3 we review the modified Rodrigues parameters, which we employ as a convenient, minimal orientation parameterization. We then describe our time delay model in Section 4. The TD-ICP algorithm is developed in Section 5. Results from simulation studies and from our laboratory calibration experiments with a PR2 robot are presented in Sections 6 and 7, respectively. We conclude in Section 8.

## 2 Related Work

Existing approaches for handling delayed measurements can be classified into three broad categories, based on a statistical characterization of the type of delay: those that handle random delays with either a known magnitude or known probability distribution, random delays with an unknown distribution, and systematic delays with an unknown magnitude. We review each in turn.

In [8], Thomopoulos and Zhang derive optimal methods for Kalman filtering with delayed measurements. Their work assumes that the delay values are known a priori, i.e., that the distributed sensor clocks are already synchronized. An alternative

<sup>2</sup> Although the orientation measurements are discrete, for smooth motion the measurements approximate a smooth curve — hence our use of the term “orientation curve.”

but inexact approach for incorporating delayed measurements in discrete-time filters is to *extrapolate* the measurements forward in time, as described by Larsen et al. in [9]. For random delays that follow a known probability distribution, Choi et al. show in [10] how an augmented-state filter can be used to maintain an extended history over the times at which each measurement could possibly have been made. By discretizing the distribution, a measurement update can be weighted and applied over all of the components of the augmented state. Finally, if the delays are known and sufficient computational resources are available, it may be possible to simply reprocess filter updates over the delay interval.

A review of control-theoretic methods for dealing with random time delays is given by Ferreira in [11]. In [12], Julier and Uhlmann describe how the *covariance union* algorithm, which was originally developed for multi-hypothesis tracking, can be used to properly incorporate measurements with random (and unknown) delays into an estimator. The result is guaranteed to be consistent with respect to possible delay values — however, covariance union is a worst-case algorithm, which typically produces an estimate with an inflated covariance.

Beyond the development of early, correlation-based methods, there has been relatively little research on systematic delay estimation for isolated, black box sensors. While substantial work has been done on clock synchronization in sensor networks (e.g., [13]), these algorithms depend on two-way communication between the sensor nodes. The devices we consider in this paper communicate in essentially one direction only (sensor-to-host), and thus the algorithms for sensor networks cannot readily be applied. One approach that is similar to our own is described by Tungadi and Kleeman in [14]. The authors determine the delay between laser range finder and odometric measurements by minimizing the hysteresis in positioning data over closed-loop trajectories. This technique does not account for noise in the laser or odometric measurements, however, while our algorithm explicitly models sensor measurement error.

### 3 The Modified Rodrigues Parameters

The modified Rodrigues parameters are most easily derived from an equivalent unit quaternion. Let  $\bar{q}$  be a unit quaternion, consisting of scalar part  $q_0$  and vector part  $\mathbf{q}$ ,

$$\bar{q} = q_0 + \mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}, \quad (1)$$

where  $q_0, q_1, q_2$  and  $q_3$  are real numbers,  $\mathbf{i}, \mathbf{j}$  and  $\mathbf{k}$  are the quaternion basis vectors, and  $\|\bar{q}\|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$ . Recall that we can interpret  $\bar{q}$  as a four-component *rotation operator*, parameterized by an angle  $\theta$  and a unit vector  $\bar{\mathbf{u}} \in \mathbb{R}^3$  [15],

$$\bar{q} \triangleq \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \bar{\mathbf{u}} \sin(\theta/2) \end{bmatrix}, \quad (2)$$

where  $\theta$  defines a rotation about the axis specified by  $\bar{\mathbf{u}}$ . The corresponding  $3 \times 1$  MRP vector,  $\rho$ , is

$$\rho = \frac{\mathbf{q}}{1 + q_0}. \quad (3)$$

Substituting (2) into (3), we obtain

$$\rho = \frac{\bar{\mathbf{u}} \sin(\theta/2)}{1 + \cos(\theta/2)} = \frac{\bar{\mathbf{u}} \sin(\theta/4)}{\cos(\theta/4)} = \bar{\mathbf{u}} \tan(\theta/4). \quad (4)$$

From (4), it is clear that the MRP vector also represents a rotation about the axis defined by  $\bar{\mathbf{u}}$ . Further,  $\rho$  is singular only when  $\theta = \pm 2\pi$  or  $\pm 360$  degrees (i.e., the singular points are as far from the origin as possible). For any rotation up to  $\pm 180$  degrees, the norm of  $\rho$  is bounded by 1. Also, for small angles,  $\rho$  linearizes as

$$\rho \approx \bar{\mathbf{u}} \frac{\theta}{4}. \quad (5)$$

We require an expression for the composition of two MRP vectors,  $\rho$  and  $\zeta$ , which defines sequential rotations by  $\zeta$  and then by  $\rho$ . In compact form, this is

$$\rho \bullet \zeta = \frac{(1 - \|\rho\|^2) \zeta + (1 - \|\zeta\|^2) \rho - 2[\zeta]_{\times} \rho}{1 + \|\rho\|^2 \|\zeta\|^2 - 2\rho^T \zeta}, \quad (6)$$

such that  $\mathbf{C}(\rho) \mathbf{C}(\zeta) = \mathbf{C}(\rho \bullet \zeta)$ , where  $\mathbf{C}(\rho)$  and  $\mathbf{C}(\zeta)$  are  $3 \times 3$  direction cosine (rotation) matrices parameterized by  $\rho$  and  $\zeta$ , respectively. The term  $[\zeta]_{\times}$  is the  $3 \times 3$  skew-symmetric cross product matrix formed from  $\zeta$ . We refer to (6) as the *rotational composition* of  $\rho$  and  $\zeta$ , and introduce the symbol ‘ $\bullet$ ’ to denote the composition operation. Finally, the MRPs have the useful property that

$$\mathbf{C}^T(\rho) = \mathbf{C}(-\rho). \quad (7)$$

## 4 System Modeling

We calibrate sensors in pairs — without loss of generality, our algorithm is developed below for one proprioceptive sensor (an IMU) and one exteroceptive sensor (a camera). Extensions to other types of proprioceptive and exteroceptive are straightforward. We model the sensor platform as a rigid body moving through space, and consider three separate reference frames at any point in time:

1. the (instantaneous) *source frame*  $\{S\}$ , which is typically associated with the sensor that has the lower update rate (e.g., the camera),
2. the (instantaneous) *target frame*  $\{D\}$ , associated with the sensor that has the higher update rate (e.g., the IMU), and

3. the *world frame*  $\{W\}$ , which serves as a fixed reference for the exteroceptive sensor.

For the experiments described herein, we define the world frame with respect to a series of corner points on a planar camera calibration pattern (cf. Figure 2).

#### 4.1 Time Delay Model

Several assumptions are necessary in order to make the calibration problem tractable. We assume that the delay is constant, i.e., that measurements from one sensor always lead or lag measurements from the other sensor by a fixed amount. We also assume that any jitter in the measurement arrival times is small enough to be ignored.

In order to compare the data from different sensors, we require a common representation — our choice here is to use orientation information. At any time  $t$ , the following continuous-time relationship holds at the receiver,

$$\rho_D^W(t) = \rho_S^W(t + \tau) \bullet \rho_D^S, \quad (8)$$

where  $\rho_D^W(t)$  is the MRP vector that defines the orientation of the target frame relative to the world frame,  $\rho_S^W(t + \tau)$  is the MRP vector that defines the (time-shifted) orientation of the source frame relative to the world frame, and  $\rho_D^S$  is the (constant but unknown) MRP vector that defines the orientation of the target frame relative to the source frame. If  $\tau$  is positive, the source measurements lag the target measurements; if  $\tau$  is negative, the source measurements lead the target measurements.

For sensors such as IMUs, which supply rate data, orientation cannot be measured directly — however, we are able to measure the *change* in orientation over a period of time, by integrating the rate values. It will be convenient to express the orientation of the target (IMU) frame relative to its initial orientation,  $\rho_D^{D_0}(t)$ , and the source (camera) frame orientation relative to the world frame. Rewriting (8), we have

$$\rho_D^{D_0}(t) = \rho_W^{D_0} \bullet \rho_S^W(t + \tau) \bullet \rho_D^S. \quad (9)$$

Since we will be dealing with discrete sequences of poses, we will identify a specific source MRP orientation vector as  $\rho_{S_i}^W$ , with instantaneous local frame  $\{S_i\}$  at time  $t_{S_i}$  (according to the receiver clock), for  $i = 1, \dots, n$  frames. Likewise, we will identify a specific target MRP orientation vector as  $\rho_{D_j}^{D_0}$ , with instantaneous local frame  $\{D_j\}$  at time  $t_{D_j}$ , for  $j = 1, \dots, m$  frames. Usually, proprioceptive sensor measurements are available more frequently and  $m \gg n$ . We call the vectors  $\rho_{S_i}^W, \rho_{D_j}^{D_0}$  *points* in the MRP orientation space. Note that  $\rho_{D_0}^{D_0} = \mathbf{0}_{3 \times 1}$ .

## 4.2 Sensor Models

We briefly review our IMU, camera, odometric and pan-tilt (encoder) sensor models below. These are the types of sensors that we calibrate on-board our PR2 robot, as described in Section 7.

### 4.2.1 IMU Model

The gyroscopes in an IMU measure angular rotation rates about three orthogonal axes. In general, IMU measurements have a bias component, i.e., the gyroscope outputs will not be zero even when the device is stationary. Gyroscope measurements are also corrupted by noise. Accounting for bias and for noise, we model the *measured* IMU angular velocity at time  $t$  as

$$\boldsymbol{\omega}_m(t) = \boldsymbol{\omega}^l(t) + \mathbf{b}_g + \mathbf{n}_g(t), \quad (10)$$

where  $\boldsymbol{\omega}^l(t)$  is the true (instantaneous) angular velocity of the IMU, resolved in the IMU frame,  $\mathbf{b}_g$  is the  $3 \times 1$  gyroscope bias vector, and  $\mathbf{n}_g(t)$  is a white Gaussian noise process with covariance matrix  $\mathbf{Q}_g(t)$  [16].

Note that the bias term above is treated as a constant. In reality, all IMUs suffer from low-frequency drift in the bias values over time. We make the assumption that the IMU biases are constant over the (short) calibration time interval, and obtain an accurate initial estimate by averaging several seconds of IMU data at the start of calibration (while the IMU is stationary).

To obtain an estimate of the IMU orientation at a time  $t > t_0$ , we integrate each IMU measurement forward according to the MRP kinematic differential equation [17, 18],

$$\dot{\boldsymbol{\rho}}(t) = \frac{1}{4} \left( \left( 1 - \|\boldsymbol{\rho}(t)\|^2 \right) \mathbf{I}_3 + 2[\boldsymbol{\rho}(t)]_{\times} + 2\boldsymbol{\rho}(t)\boldsymbol{\rho}^T(t) \right) \boldsymbol{\omega}^l(t), \quad (11)$$

where  $\mathbf{I}_3$  is the  $3 \times 3$  identity matrix.

As we continue integrating, we incorporate inaccuracies due to noise and (possibly) drift, and so become less certain about the true orientation of the IMU relative to its initial orientation. We maintain an estimate of this uncertainty by propagating the IMU orientation covariance forward in time via numerical integration.

### 4.2.2 Camera Model

We use an ideal projective (pinhole) camera model, and assume that the camera intrinsic and lens distortion parameters are known. Each camera measurement  $\mathbf{m}_{ik}$  is the projection of corner point  $l_k$  on the calibration pattern, at position  $\mathbf{p}_{ik}^{C_i} = [x_{ik} \ y_{ik} \ z_{ik}]^T$  in camera frame  $\{C_i\}$ , onto the image plane,

$$\mathbf{m}_{ik} = \begin{bmatrix} u_{ik} \\ v_{ik} \end{bmatrix} = \begin{bmatrix} x'_{ik} \\ y'_{ik} \end{bmatrix} + \boldsymbol{\eta}_{ik}, \quad \begin{bmatrix} x'_{ik} \\ y'_{ik} \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} x_{ik}/z_{ik} \\ y_{ik}/z_{ik} \\ 1 \end{bmatrix}, \quad (12)$$

where  $[u_{ik} \ v_{ik}]^T$  is the vector of observed image coordinates,  $\mathbf{K}$  is the  $3 \times 3$  camera intrinsic calibration matrix [19], and  $\boldsymbol{\eta}_{ik}$  is a zero-mean Gaussian measurement noise vector with covariance matrix  $\mathbf{R}_{ik} = \sigma_{ik}^2 \mathbf{I}_2$ .

From the observations of several known corner points on the calibration pattern, we obtain an estimate of the camera orientation in the world frame (and the associated covariance matrix) using a nonlinear least squares computation. The computation requires an initial approximate orientation for the camera, which we determine using Ansar’s algorithm [20].

### 4.2.3 Odometric and Pan-Tilt Sensor Models

Odometric and pan-tilt sensors provide orientation information directly, in the form of wheel encoder and joint encoder measurements, respectively. Note that a pan-tilt unit has only two degrees of rotational freedom, while a mobile base has only one degree of rotational freedom (i.e., yaw). To compare these measurements with the three degrees-of-freedom camera or IMU data, we set the “extra” components of the MRP orientation vectors (corresponding to the non-actuated degrees of freedom) to zero.

## 5 Time Delay Calibration

In this section, we present the TD-ICP algorithm; pseudo-code is listed in Algorithm 1. The algorithm initially generates  $n$  source MRP orientation vectors (cf. Section 4.2.2) from, e.g., captured images of a calibration pattern. It then computes a set of  $m$  target MRP orientation vectors (cf. Section 4.2.1) by, e.g., integrating IMU data forward in time.

To estimate the source-target relative orientation and the time delay, the algorithm iteratively chooses  $n$  point correspondences (assuming  $n < m$ ) and computes the spatial and temporal transforms between the orientation curves. As the curves converge, the time range over which the correspondence search is carried out (for each source point) is adjusted. These steps are described in more detail below.

### 5.1 A Registration Metric

TD-ICP operates by iteratively selecting, for each point on the source orientation curve, the closest neighboring point on the target curve. The notion of “closeness”

**Algorithm 1** Time Delay Iterative Closest Point (TD-ICP)

- 
- 1: **for each** exteroceptive sensor measurement, compute  $\rho_{S_i}^W$  at time  $t_{S_i}$
  - 2: **for each** proprioceptive sensor measurement, compute  $\rho_{D_j}^{D_0}$  at time  $t_{D_j}$ , possibly using  $\rho_{D_{j-1}}^{D_0}$
  - 3: initialize  $\hat{\rho}_W^{D_0}$  and  $\hat{\rho}_D^S$  with coarse relative orientation estimates
  - 4: initialize  $\hat{\tau} = 0$
  - 5: **repeat**
  - 6:   **for each** point  $\hat{\rho}_W^{D_0} \bullet \rho_{S_i}^W \bullet \hat{\rho}_D^S$ , find closest point  $\rho_{D_j}^{D_0}$  in timestamp range  $[t_{S_i} - \hat{\tau} - \delta t, t_{S_i} - \hat{\tau} + \delta t]$
  - 7:   compute new estimates of  $\hat{\rho}_W^{D_0}$  and  $\hat{\rho}_D^S$  using iterated nonlinear total least squares
  - 8:   compute new  $\hat{\tau} = \sum (t_{S_i} - t_{D_j}) / n$  from  $n \{i, j\}$  pairs
  - 9: **until**  $\hat{\rho}_W^{D_0}$ ,  $\hat{\rho}_D^S$  and  $\hat{\tau}$  have converged
- 

requires a suitable distance metric — our metric is based on arc length on the unit sphere. For each source point  $\rho_{S_i}^W$ , we determine the distance to a nearby target point  $\rho_{D_j}^{D_0}$  by computing the incremental MRP vector that takes  $\rho_{S_i}^W$  to the same orientation as  $\rho_{D_j}^{D_0}$ , after applying the current best estimate of the spatial transform<sup>3</sup>,

$$\sigma_{ij} = \left( - \left( \hat{\rho}_W^{D_0} \bullet \rho_{S_i}^W \bullet \hat{\rho}_D^S \right) \right) \bullet \rho_{D_j}^{D_0}. \quad (13)$$

The closest point is the target point that has the smallest value for the distance function

$$d_{ij} = 4 \arctan \left( \sqrt{\sigma_{ij}^T \sigma_{ij}} \right), \quad (14)$$

which is the arc length, in radians, of the incremental rotation that takes  $\rho_{S_i}^W$  (with the transform applied) to  $\rho_{D_j}^{D_0}$  (see (4) in Section 3). This function satisfies the four required properties of a metric [21].

For each source point, the search range on the target curve is bounded by the maximum anticipated time delay. This range is, explicitly,  $[t_{S_i} - \hat{\tau} - \delta t, t_{S_i} - \hat{\tau} + \delta t]$ , where  $\pm \delta t$  is the maximum delay value.

## 5.2 Iterative Nonlinear Registration

After finding the closest target point for each source point, we have a set of  $n$  correspondences between the orientation curves. Let these  $n \{i, j\}$  pairs identify the matching source and target reference frames,  $\{S_i\}$  and  $\{D_j\}$ , respectively. We define a mapping function  $f(i)$  which returns the corresponding target frame  $\{D_j\}$  for source frame  $\{S_i\}$ . The curves are aligned using a generalized nonlinear total least squares computation, in which we minimize the following cost function,

---

<sup>3</sup> We use the ‘^’ (hat) notation throughout the paper to denote an estimated quantity.

$$U(\hat{\rho}_W^{D_0}, \hat{\rho}_D^S) = \sum_{i=1}^n \mathbf{s}_i^T \mathbf{P}_{S_i}^{-1} \mathbf{s}_i + \sum_{i=1}^n \mathbf{t}_i^T \mathbf{P}_{D_{f(i)}}^{-1} \mathbf{t}_i, \quad (15)$$

over the transform parameters,  $\hat{\rho}_W^{D_0}$  and  $\hat{\rho}_D^S$ . On the right-hand side of (15),

$$\mathbf{s}_i = \rho_{S_i}^W - \hat{\rho}_{S_i}^W, \quad (16)$$

$$\mathbf{t}_i = \rho_{D_{f(i)}}^{D_0} - \hat{\rho}_{D_{f(i)}}^{D_0}, \quad (17)$$

are the residuals in  $\hat{\rho}_{S_i}^W$  and  $\hat{\rho}_{D_{f(i)}}^{D_0}$ , respectively, while  $\mathbf{P}_{S_i}$  and  $\mathbf{P}_{D_{f(i)}}$  are the associated covariance matrices. Note that (15) is the sum of squared Mahalanobis distances between the computed points and measured points on the source and target curves; we consider the uncertainty in both sets of measurements. The residuals are subject to the  $n$  constraints

$$\hat{\rho}_{D_{f(i)}}^{D_0} - \hat{\rho}_W^{D_0} \bullet \hat{\rho}_{S_i}^W \bullet \hat{\rho}_D^S = 0, \quad i = 1, \dots, n. \quad (18)$$

It is possible to minimize (15), subject to the constraints of the form defined by (18), with the use of Lagrange multipliers [22]. Incorporating the constraints, differentiating and rearranging, we obtain

$$\begin{bmatrix} \Delta \rho_W^{D_0} \\ \Delta \rho_D^S \end{bmatrix} = \left[ \sum_{i=1}^n (\mathbf{J}_i^T \mathbf{M}_i^{-1} \mathbf{J}_i) \right]^{-1} \left[ \sum_{i=1}^n \mathbf{J}_i^T \mathbf{M}_i^{-1} \Delta \mathbf{y}_i \right] \quad (19)$$

where  $\Delta \rho_W^{D_0}$  and  $\Delta \rho_D^S$  are the incremental updates to  $\hat{\rho}_W^{D_0}$  and  $\hat{\rho}_D^S$ , respectively, for the current iteration,  $\mathbf{M}_i = \mathbf{P}_{D_{f(i)}} + \mathbf{H}_{S_i} \mathbf{P}_{S_i} \mathbf{H}_{S_i}^T$ ,  $\Delta \mathbf{y}_i = \rho_{D_{f(i)}}^{D_0} - \hat{\rho}_{D_{f(i)}}^{D_0}$ , and  $\mathbf{J}_i$  is the Jacobian with respect to the transform parameters,

$$\mathbf{J}_i(\hat{\rho}_W^{D_0}, \hat{\rho}_D^S) = \left[ \begin{array}{c} \frac{\partial(\rho_W^{D_0} \bullet \rho_{S_i}^W \bullet \rho_D^S)}{\partial \rho_W^{D_0}} \Big|_{\hat{\rho}_W^{D_0}, \hat{\rho}_D^S} \\ \frac{\partial(\rho_W^{D_0} \bullet \rho_{S_i}^W \bullet \rho_D^S)}{\partial \rho_D^S} \Big|_{\hat{\rho}_W^{D_0}, \hat{\rho}_D^S} \end{array} \right]. \quad (20)$$

We omit the full Jacobian expression for brevity (the Jacobian is complicated but straightforward to evaluate numerically). From the correspondences, we can also immediately update the estimate of  $\tau$  as

$$\hat{\tau} = \frac{\sum_{i=1}^n (t_{S_i} - t_{D_{f(i)}})}{n} \quad (21)$$

for the  $n$  pairs. The closest point selection and nonlinear alignment steps are iterated to convergence. This yields values for the time delay and the source-target relative orientation.

The alignment computation requires an initial estimate of the relative orientation of the sensors. This estimate can be fairly coarse, and we use simple hand measurement of the relative orientation for our work here.

## 6 Simulation Studies

We initially evaluated the performance of the TD-ICP algorithm through a series of Monte Carlo simulations of camera-IMU calibration. The characteristics of the simulated camera and IMU were the same as those of hardware available in our laboratory.

We chose a set of three different sensor platform trajectories, with the camera and the IMU rotating and translating in front of a simulated planar calibration pattern. For each trajectory, we ran 100 simulations, with a different time delay and a different orientation perturbation at the start of each trial. The random time delay values were drawn from a zero-mean Gaussian distribution with a standard deviation of 25 ms; the initial orientation perturbations were drawn from a zero-mean multivariate Gaussian distribution with a standard deviation of  $5^\circ$  in roll, pitch and yaw. In all cases, the true relative orientation of the IMU with respect to the camera was the same,  $\rho_D^s = -[0.33 \ 0.33 \ 0.33]^T$ .

The motion durations for the trajectories were 5, 6, and 8 seconds (Trajectory 1 to Trajectory 3, respectively). Simulated camera images were captured at 15 frames per second, while IMU updates occurred at 100 Hz. For each image, we projected the corner points of the calibration pattern onto the camera image plane, and added independent, zero-mean Gaussian noise to the  $u$  (horizontal) and  $v$  (vertical) pixel coordinates. The noise had a standard deviation of 0.5 pixels in  $u$  and  $v$ . When generating the IMU measurements, we also added zero-mean Gaussian noise to the angular rate values.

Simulation results are presented in Table 1. We note that the mean delay error for all three trajectories is approximately zero, indicating that the estimation process is unbiased. Further, the mean absolute error in the estimate of  $\tau$  ranges from less than 1 ms for Trajectory 2, to 1.4 ms for Trajectory 1. The maximum absolute delay error is 5.7 ms, which is less than the time between IMU updates.

An example of the initial and final alignment of the orientation curves for one of the trials from Set 3 is shown in Figures 1. In this case, the error in the value of  $\tau$  is 0.591 ms. The final rotation error is  $0.04^\circ$ .

To verify that the final delay errors were not correlated with the initial delay values, we computed the correlation coefficients between the initial delay and the final

Table 1: Monte Carlo simulation results. Each set includes 100 simulation trials, with a different time delay and a different orientation perturbation for each trial.

Set	Avg. Iters.	$\tau$ Error (ms)				$\rho_D^s$ Error ( $^\circ$ )	
		Mean	Stdev.	Mean Abs.	Max. Abs	Mean Abs.	Stdev.
1	20.4	-0.04	1.86	1.40	5.67	0.44	0.21
2	20.4	0.29	0.99	0.84	3.23	0.37	0.18
3	18.7	0.44	1.38	1.15	3.67	0.29	0.12

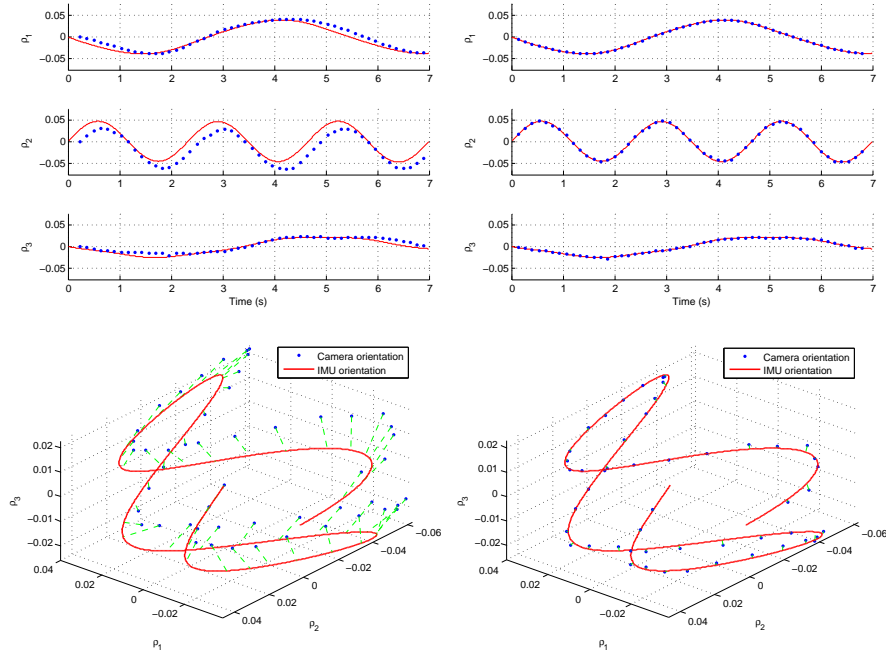


Fig. 1: Simulation results for Trajectory 3. The left column shows the estimated orientation of the IMU (red line) and camera (blue dots) over time, prior to calibration. The right column shows the time-shifted camera curve after the TD-ICP algorithm has converged. Green lines indicate initial point correspondences, selected according to our distance metric. The true time delay was 150 ms in this case; the error in the final delay estimate is 0.59 ms.

delay error, for each trajectory. The correlation coefficients were  $-0.1503$ ,  $0.1889$  and  $-0.0008$  for Sets 1 to 3, respectively. None of these correlations were statistically significant (for significance level  $\alpha = 0.05$ ).

## 7 Experiments with the PR2 Robot

In order to determine the performance of the TD-ICP algorithm using data from real hardware, we calibrated four of the sensors on-board USC's PR2 robot. The sensors included the left wide-angle stereo camera, head pan-tilt unit, base odometer and the torso IMU. The locations of the individual sensors (relative to the body of the PR2) are shown in Figure 2. All of the proprioceptive sensors (pan-tilt, odometer, IMU) were calibrated with respect to the exteroceptive sensor (the left stereo camera, in this case).

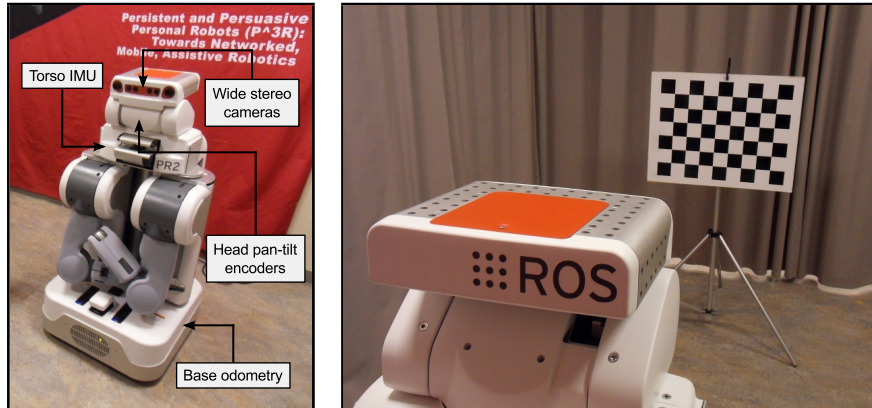


Fig. 2: Left: to demonstrate the TD-ICP algorithm, we cross-calibrated the left wide-angle stereo camera, head pan-tilt encoders, base odometry encoders and the torso IMU. Right: PR2’s-eye view of the planar calibration pattern used as a reference for the camera.

The calibration procedure involved first locking the head pan-tilt unit in place and rotating the entire PR2 slowly to the left and to the right. This provided data for camera-IMU and camera-odometry calibration. We then moved the head with the pan-tilt motors, while keeping the base stationary and ensuring that the planar calibration pattern remained within the camera’s field of view. When calculating the time delays using TD-ICP, we rejected any source-target matches with distances greater than three times the standard deviation of all distance values [23].

Calibration results are shown in Table 2. These results are for single trials only, however in all cases we obtained delay values that were within 2 ms of the values in the table, over at least three trials.

Table 2: Calibration results for multiple sensors on the PR2 robot. The time delay and orientation values for the three proprioceptive sensors (head pan-tilt unit, base odometer and torso IMU) are specified relative to the single exteroceptive sensor (left wide-angle stereo camera).

Sensor	Motion Dur. (s)	TD-ICP Iters.	Camera-Relative Calibration Parameters			
			$\tau$ (ms)	Roll ( $^{\circ}$ )	Pitch ( $^{\circ}$ )	Yaw ( $^{\circ}$ )
Pan-Tilt	7.1	102	-28.0	-90.31	-0.09	-89.13
Odometry	8.1	76	-10.0	-98.59	-3.94	-87.28
IMU	8.0	31	1.5	80.83	-2.75	-86.67

## 8 Conclusions and Ongoing Work

This paper presented an algorithm, TD-ICP, for accurate temporal calibration of multiple proprioceptive and exteroceptive sensors. We have shown that it is possible to treat temporal calibration as a *registration* problem. The ICP-based approach requires only a small amount of data and demonstrates fast and reliable convergence. Our results indicate that it is possible to calibrate the relative time delay between sensor data streams with accuracies on the order of 1 to 2 ms.

As ongoing work, we are exploring the use of TD-ICP to calibrate the time delays in other multi-sensor systems. For example, we are investigating the use of this method to determine the relative time delay between measurements from inertial and LIDAR sensors.

## Acknowledgments

This work was funded in part by the US NSF (grants IIS-1017134 and CCF-0120778), by DARPA under the Autonomous Robot Manipulation (ARM) program (contract W91CRBG-10-C-0136), by Willow Garage under the PR2 Beta program and by a gift from the Okawa Foundation. Jonathan Kelly was supported by an Anenberg Fellowship from the University of Southern California.

## References

1. D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic Self Calibration of a Camera and a 3D Laser Range Finder from Natural Scenes," in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS'07)*, San Diego, USA, Oct./Nov. 2007, pp. 4164–4169.
2. F. M. Mirzaei and S. I. Roumeliotis, "A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1143–1156, Oct. 2008.
3. J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-Calibration," *Int'l J. Robotics Research*, vol. 30, no. 1, pp. 56–79, Jan. 2011.
4. C. N. Knapp and G. C. Carter, "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-24, no. 4, pp. 320–327, Aug. 1976.
5. Y. T. Chan, J. M. F. Riley, and J. B. Plant, "A Parameter Estimation Approach to Time-Delay Estimation and Signal Detection," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-28, no. 1, pp. 8–16, Feb. 1980.
6. Y. Chen and G. Medioni, "Object Modeling by Registration of Multiple Range Images," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA'91)*, vol. 3, Sacramento, USA, Apr. 1991, pp. 2724–2729.
7. P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
8. S. C. A. Thomopoulos and L. Zhang, "Decentralized Filtering with Random Sampling and Delay," *Information Sciences*, vol. 81, no. 1-2, pp. 117–131, Nov. 1994.

9. T. D. Larsen, N. A. Andersen, O. Ravn, and N. K. Poulsen, "Incorporation of Time Delayed Measurements in a Discrete-time Kalman Filter," in *Proc. IEEE Conf. Decision & Control (CDC'98)*, vol. 4, Tampa, USA, Dec. 1998, pp. 3972–3977.
10. M. Choi, J. Choi, J. Park, and W. K. Chung, "State Estimation with Delayed Measurements Considering Uncertainty of Time Delay," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA'09)*, Kobe, Japan, May 2009, pp. 3987–3992.
11. A. R. Ferreira and J. M. Fernandes, "A Survey on Time Delay System Estimation," in *Proc. European Control Conf. (ECC'97)*, Brussels, Belgium, July 1997.
12. S. J. Julier and J. K. Uhlmann, "Fusion of Time Delayed Measurements with Uncertain Time Delays," in *Proc. American Control Conf. (ACC'05)*, vol. 6, Portland, USA, June 2005, pp. 4028–4033.
13. Q. Li and D. Rus, "Global Clock Synchronization in Sensor Networks," *IEEE Trans. Computers*, vol. 55, no. 2, pp. 214–226, Feb. 2006.
14. F. Tungadi and L. Kleeman, "Time Synchronisation and Calibration of Odometry and Range Sensors for High-Speed Mobile Robot Mapping," in *Proc. Australasian Conf. Robotics and Automation (ACRA'08)*, J. Kim and R. Mahony, Eds., Canberra, Australia, Dec. 2008.
15. J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton, USA: Princeton University Press, Aug. 2002.
16. A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*, ser. Progress in Astronautics and Aeronautics, P. Zarchan, Ed. Cambridge, USA: American Institute of Aeronautics and Astronautics, Sept. 1997, vol. 174.
17. J. L. Crassidis and F. L. Markely, "Attitude Estimation Using Modified Rodrigues Parameters," in *Proc. Flight Mechanics and Estimation Theory Symp.* Greenbelt, USA: NASA Goddard Space Flight Center, May 1996, pp. 71–83.
18. H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*, 2nd ed., ser. AIAA Education Series, J. A. Schetz, Ed. Reston, USA: American Institute of Aeronautics and Astronautics, Sept. 2009.
19. Y. Ma, S. Soatto, J. Košecká, and S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, 1st ed., ser. Interdisciplinary Applied Mathematics. New York: Springer, Nov. 2004, vol. 26.
20. A. Ansar and K. Daniilidis, "Linear Pose Estimation from Points or Lines," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 578–589, May 2003.
21. D. Q. Huynh, "Metrics for 3D Rotations: Comparison and Analysis," *J. Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, Oct. 2009.
22. P. Gans, *Data Fitting in the Chemical Sciences: By the Method of Least Squares*. Chichester, United Kingdom: Wiley, Nov. 1992.
23. S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm," in *Proc. 3rd Int'l Conf. 3-D Digital Imaging and Modeling*, Québec City, Canada, May/June 2001, pp. 145–152.