# Evaluation of Combined Time-Offset Estimation and Hand-Eye Calibration on Robotic Datasets

Fadri Furrer, Marius Fehr, Tonci Novkovic, Hannes Sommer, Igor Gilitschenski, and Roland Siegwart

**Abstract** Using multiple sensors often requires the knowledge of static transformations between those sensors. If these transformations are unknown, hand-eye calibration is used to obtain them. Additionally, sensors are often unsynchronized, thus requiring time-alignment of measurements. This alignment can further be hindered by having sensors that fail at providing useful data over a certain time period. We present an end-to-end calibration framework to solve the hand-eye calibration. After an initial time-alignment step, we use the time-aligned pose estimates to perform the static transformation estimation based on different prefiltering methods, which are robust to outliers. In a final step, we employ a non-linear optimization to locally refine the calibration and time-alignment. Successful application of this estimation framework is demonstrated on multiple robotic systems with different sensor configurations. This framework is released as open source software together with the datasets.

## 1 Introduction

The hand-eye calibration problem is among the most important calibration scenarios in robotics. Its name refers to the problem of calibrating the pose of a camera coordinate system relative to the reference frame of the robot arm's end effector on which it is rigidly mounted. Another important instance of the problem is inferring the relative pose of two sensors, such as cameras, even if their views do not overlap. More generally, hand-eye calibration systems aim at finding the transformation between two reference frames that are rigidly mounted with respect to each other.

Formally, solving the hand-eye calibration problem comes down to solving the $AX = XB$ equation in which $A$, $B$, and $X$ represent rigid body motions. This formu-

Autonomous Systems Lab, ETH Zurich, Leonhardstrasse 21, 8092 Zurich, Switzerland, e-mails: { fadri.furrer, marius.fehr, tonci.novkovic, hannes.sommer }@mavt.ethz.ch, { igilitschenski, rsiegwart }@ethz.ch

lation, originally proposed in [21], has been subject of an extensive body of research which focused on finding a solution to this equation. However, practical implementations of a hand-eye calibration system may present significant additional challenges. For instance, time-alignment needs to be taken into account when the two reference frames stem from sensors / actuators running on different systems. This is particularly true when these systems need to be (re-)calibrated on-line during regular operation and, therefore, cannot be specifically controlled for calibration.

For practical applications it is of particular interest to be able to solve the aforementioned problems within a single system making it widely applicable to different practical instances of the hand-eye calibration problem. Unfortunately, the broad body of research on the hand-eye calibration problem is not adequately matched by thorough evaluations in different scenarios and freely available software packages.

The goal of this work is to fill this gap by providing an open source toolbox[1] for hand-eye calibration that can be easily used within a broad range of applications and is at the same time easily adaptable to incorporating further algorithms and calibration procedures. Contributions presented in this paper not only involve the presentation of the software package but also its applicability to robotic systems. This is achieved through thorough evaluations on different types of datasets involving a robotic arm and multiple hand-held devices. Furthermore, we make all our datasets publicly available in order to simplify future evaluation of hand-eye calibration algorithms. The contributions of this work can be summarized as follows:

- A collection of datasets using different sensors and sensor configurations.
- Thorough validation of the hand-eye calibration system with different filtering methods on these datasets.
- A software toolbox for hand-eye calibration including time-alignment and handling noisy data.

The remainder of the paper is structured as follows. In the next section, we study related work. An overview of the methodology implemented in the proposed calibration toolbox is presented in Section 3. The datasets that are used for evaluation are presented in Section 4 followed by the validation of the proposed method. A conclusion with an outlook is provided in Section 6.

## 2 Related Work

The problem of hand-eye calibration has been well studied in late 80's and 90's. Classical approaches to solving $AX = XB$ problem decoupled rotational and translational parts of the calibration, resulting in simpler but more error-prone solutions [23]. Shiu and Ahmad [21] demonstrated how a hand-eye calibration problem can be expressed using an angle-axis representation and solved for rotation, then translation using a least squares fitting. Similar, but a more efficient approach was

[1] `https://github.com/ethz-asl/hand_eye_calibration`

developed by Tsai and Lenz [25] using a closed-form solution. Wang [26] proposed another formulation using angle-axis representation and conducted an early comparison of the three methods, reporting that the one by Tsai and Lenz [25] performed the best on average. Formulation of the same problem using quaternions for rotations was introduced by Chou and Kamel [3]. Park and Martin [18] have formed an alternative closed-form solution using Lie group theory to simplify the problem, and Fassi and Legnani [6] demonstrated how to solve the calibration problem, in a least squares manner, first for rotation and then translation in presence of noisy data.

The same $AX = XB$ problem can be solved simultaneously for hand-eye rotation and translation. Horaud and Dornaika [13], in addition to proposing another closed-form solution using quaternions for the decoupled problem, also proposed an iterative method for solving the orientation (represented by quaternions) and translation components simultaneously. They applied a Levenberg-Marquardt technique, a robust non-linear optimization method, to obtain the solution. Furthermore, they performed a stability analysis for both of their approaches and the method proposed by Tsai and Lenz [25], concluding that the non-linear optimization method is the most robust with respect to measurement errors and noise, and much more accurate than the classical formulation by Tsai and Lenz [25]. Daniilidis [4] proposed another formulation, based on screw-theory, for the simultaneous hand-eye calibration. He obtained a singular value decomposition (SVD)-based solution by using a dual-quaternion representation for both rotations and translations. His work was extended by Schmidt et al. [20] who also implemented the screw-axis based selection of movement pairs for increasing numerical stability and random sample consensus (RANSAC)-based elimination of outliers. Another iterative method based on a parameterization of a stochastic model was introduced by Strobl and Hirzinger [23]. In order to evaluate optimality of different algorithms, they introduced a metric on the group of the rigid transformations SE(3) and the corresponding error model for non-linear optimization.

Andreff and Espiau [2] demonstrated robot hand-eye calibration using structure-from-motion for computing camera motions, up to an unknown scale factor which is introduced in a linear formulation of the calibration problem. They also showed that their method is very accurate in rotation, however, for translations, in case of noisy data, other methods by Daniilidis [4] and Horaud and Dornaika [13] perform better. A modification to the structure-from-motion approach was presented by Heller et. al. [12] which addresses the scale ambiguity by formulating the estimation of the hand-eye displacement as an $L_\infty$-norm optimization problem.

In most practical applications, in addition to estimating the hand-eye calibration, and due to asynchronous clocks from different devices, it is necessary to perform temporal alignment of the data. Ackerman et. al. [1] used invariant quantities, coming from screw theory, between two pairs of measurements to align uniformly asynchronous data and account for data with gaps. Alignment was based on correlation of the measurement invariants using the Discrete Fourier Transform (DFT), however, the approach was evaluated only on simulated data. In their motion-based calibration method, Taylor and Nieto [24] compute the likelihood of a timing offset based on an angle through which each sensor rotates, taking the associated uncer-

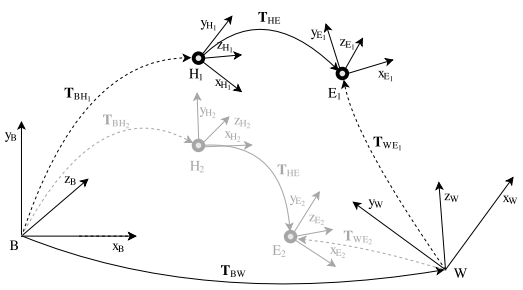


**Fig. 1** The transformations relevant for the and-eye calibration. Black are the transformations of the first pose pair and in grey the second pose pairs. Solid lines indicate static transformations, where dashed lines indicate transformations that change over time.

tainty into account. Additionally, using a probabilistic framework and based on the estimated motion of each individual sensor, estimated accuracy of each sensor's readings and appearance information, they compute the final calibration. Rehder et. al. [19] have demonstrated a general framework, using a continuous-time state representation, for joint calibration of temporal offsets and spatial transformations between multiple sensors. In this approach, the time offset is estimated using basis functions which allows them to treat the problem within the rigorous theoretical framework of maximum likelihood estimation. An alternative approach, formulating the temporal calibration as a registration task, using an iterative closest point (ICP) algorithm, was introduced by Kelly and Sukhatme [14]. TICSync, an open source library implementing software for time-alignment was developed by Harrison and Newman [11]. They used a two-way timing mechanism to estimate the offset and realize unified and precise timing across distributed networked systems. Since sensors rarely have support for this two-way mechanism, another open-source framework, TriggerSync by English et. al. [5] was developed based on TICSync library. This framework is used for synchronizing multiple triggered sensors with respect to the local clock.

Our approach is based on the method by Daniilidis [4] with a similar outlier rejection and motion selection as in Schmidt et al. [20]. However, our method incorporates several additional outlier rejection techniques, of which we prove that they can significantly improve the performance of the original algorithm. Furthermore, we perform initial time-alignment based on correlation between the angular velocities. As a final refinement step we perform non-linear maximum likelihood batch estimation with a continuous-time state representation as described in [9]. The overall approach for the this step is very similar to the one proposed in [19].

## 3 Method

In the presented work, we allow inputs to the hand-eye calibration to be pose estimations or camera images from which we estimate poses relative to a visual target. Therefore, we split this section into the subsections for target extraction, time-alignment, hand-eye calibration, and the refinement step. The pose estimations from

camera images are described in Section 3.1. We use interpolated angular velocity norms for the time-alignment, which we describe in Section 3.2. With two timely aligned sets of poses, we perform the hand-eye calibration with outlier rejection, as described in Section 3.3. Using the results from time-alignment and (global) hand-eye calibration as initial guesses, we additionally perform a final local refinement step using non-linear optimization to find a local, joint spatiotemporal maximum likelihood solution. This last step we describe in Section 3.3.2.

## *3.1 Target Extractor*

In order to use the time-alignment and hand-eye calibration methods presented in the following sections, two sets of poses are required. There are several well known methods to estimate poses from camera images, based on feature matching, optical flow, etc. To avoid drift in the measurements it is beneficial to find features of an object that is known to be stationary in the environment. Camera pose estimates from matched features suffer from an unknown scale, in order to solve this issue, one can look for pairs of features with known metric distances, or use additional sensors with metric information, such as inertial measurement units, range sensors, radar, motor, or wheel encoders.

In our approach, we use visual *AprilTag* targets [17] of known size. We assume that the intrinsic camera calibrations are known[2]. When the target is visible in the camera frame, corner features are extracted. Additionally, by detecting *AprilTags* on the calibration target, the detected corners can produce one to one matches among different images. The successful observations of the target are appended to a vector. For all these observations we check, using a RANSAC based Perspective-n-Point method, if they agree with the camera model from the intrinsic calibration and extract a pose estimation of the camera (or eye) $E$, $\mathbf{T}_{WEi}$, in the target (or world) frame $W$. If the corresponding inlier ratio $\lambda_i$ is greater then an inlier threshold $\lambda_{th}$ we keep the pose estimate $\mathbf{T}_{WEi}$ along with its timestamp $t_{Ei}$ for the next steps.

## *3.2 Time Alignment*

In order to compare poses originating from different sensors one can rarely rely on hardware-synchronized device clocks as the sensors might not be communicating at all, e.g. when calibrating a camera tracked by an external motion capture system. That is why the synchronization of the two sensor clocks, or to be more precise the two sets of timestamped sensor data is the first crucial step for hand-eye calibration. A popular method of computing the time-alignment for signals with constant time-offsets is to correlate the angular velocity norms of both pose signals. To that end,

[2] For intrinsic camera calibration, the Kalibr framework (`https://github.com/ethz-asl/kalibr`) was used and we refer the reader to [7, 8, 16] for more details.

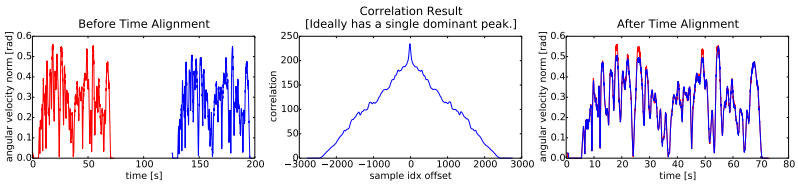


Fig. 2: Time-alignment result: The plot provides an intuitive understanding of the direction and magnitude as well as the quality of the alignment.

we first resample the poses at the lower frequency of the two pose signals and then compute the angular velocity norm based on both sets of quaternions. In order to make the time-alignment more robust to outliers or missing data, e.g. caused by a signal drop, we first cap the signal at the $99th$ percentile of the magnitude and then apply a low-pass kernel. The time offset can then be computed from the maximum value of the convoluted signal. In order to provide feedback about the quality of the time-alignment, the user is presented a comprehensible graphical representation of the alignment results (see Figure 2).

## *3.3 Hand-Eye Calibration*

To perform a hand-eye calibration, at least two pose pairs are required. As depicted in Figure 1, we can then solve the hand-eye calibration equation:

$$\mathbf{T}_{BH_1}\mathbf{T}_{HE}\mathbf{T}_{WE_1}^{-1} = \mathbf{T}_{BH_2}\mathbf{T}_{HE}\mathbf{T}_{WE_2}^{-1}\,, \tag{1}$$

where $B$ is the body frame, $H$ the hand frame, $W$ and $E$, the world and eye frame introduced earlier. This can be reformulated using the transformation between consecutive poses of the two pose sources, using $\mathbf{T}_{H_1H_2} = \mathbf{T}_{BH_2}^{-1}\mathbf{T}_{BH_1}$ , and $\mathbf{T}_{E_1E_2} = \mathbf{T}_{WE_2}^{-1}\mathbf{T}_{WE_1}$ , respectively, into $\mathbf{T}_{H_1H_2}\mathbf{T}_{HE} = \mathbf{T}_{HE}\mathbf{T}_{E_1E_2}$.

In the context of this paper, the method presented in [4] was used for the hand-eye calibration. Therefore, we are solving the hand-eye calibration using dual quaternions $\check{\mathbf{q}}_{H_1H_2} = \check{\mathbf{q}}_{HE}\check{\mathbf{q}}_{E_1E_2}\check{\mathbf{q}}_{HE}^{-1}$. However, this method is sensitive to outlier and noise as it employs an SVD to solve the hand-eye calibration problem.

### 3.3.1 Filtering

To improve the robustness and the accuracy of the calibration results, we implemented and evaluated several outlier rejection and filtering methods. First, in order to reduce the amount of data points we need to process, we employ the following filtering technique:

**Pose Filtering (PF)** Since usual datasets can contain very large number of pose pairs for calibration, in the first step of our approach, we apply a filtering method based on [20]. This method first computes the screw motion axis of each dual quaternion representing one transformation. The dot products for each combination of the screw-axis from the hand data, as well as the dot products of each combination of the eye data are computed. If one of these dot products is higher than a threshold then the respective hand-eye pair gets filtered out. The main idea for this filtering is that if the dot product is high, it means that the screw-axis for these transforms are almost parallel, meaning that they contain similar information and that we can filter one of them out since it is not so informative. Using this filtering method we can greatly reduce the number of pose pairs that will be passed to the calibration algorithm, thus, improving the efficiency, however, slightly reducing the accuracy.

**RANSAC "Classic" (RC)** In a first, step we reduce the number of dual quaternion pairs using the filtering method described above. RANSAC (see Algorithm 1) draws $n$ ($n \geq 2$) time-aligned quaternion pairs at random, the *sample*. As described in [4], the scalar parts of two dual quaternions representing the same screw need to be equal in order for this method to succeed. We made use of this condition to first reject any *samples* that violate it early on. RANSAC then proceeds by identifying inliers that agree with the hand-eye calibration. Therefore, the standard way is to first estimate the calibration based on the drawn *samples*. This resulting calibration is then used to transform the quaternion pairs into the same base frame. We then compare their position and orientation errors which allows us to apply thresholds $\lambda_{t,min}$ (position) and $\lambda_{r,min}$ (orientation) to identify inliers and outliers. The calibration is refined by repeating the hand-eye calibration method on the inliers found in the previous step. We repeat the evaluation step we used to identify the inliers and compute the root mean square error (RMSE) of position and orientation across all the quaternion pairs. Finally, we keep the calibration that exhibits the lowest RMSE.

**RANSAC Scalar (RS) based inlier check** Furthermore, we propose and compare a second variant of this algorithm that employs a different, more efficient way of identifying inliers. We reduce the *sample* size to 1 and omit the *sample*-based hand-eye calibration computation and its expensive evaluation, and directly select the inliers based on the compatibility of the quaternion pairs' scalar values. The algorithm then continues like the previous RANSAC variant by computing the calibration on the inliers and evaluating it based on the RMSE of position and rotation error of the aligned quaternion pairs.

We compare our proposed improvements to the following two algorithms.

- **Baseline (B):** Finds the first subset of quaternion pairs that fulfills the scalar value equality condition and compute the hand-eye-calibration.
- **Exhaustive search (EC and ES):** Is algorithmically identical to the proposed RANSAC algorithms, except that all possible sample combinations are explored. In order to keep the runtime within reasonable limits we employ this method only on the filtered set of quaternion pairs.

**Algorithm 1:** RANSAC based input pose pair selection for Eq. 1.

**Data**: A pair of vectors with time-aligned dual quaternions: $\mathbf{P}_{a,b} = [a,b]$
$a = [\check{\mathbf{q}}_{a,1} \cdots \check{\mathbf{q}}_{a,k}]^T, \quad b = [\check{\mathbf{q}}_{b,1} \cdots \check{\mathbf{q}}_{b,k}]^T, \quad RMSE_{best} = \infty$
**Result**: Static transform dual quaternion $\check{\mathbf{q}}_{a,b}$ and corresponding RMSE
**Function** HandEyeCalibrationRANSAC($\mathbf{P}_{a,b}$)
  $\mathbf{F}_{a,b} \leftarrow$ FilterPairs($\mathbf{P}_{a,b}$) **// PF**
  **while** *not reached probability of at least one inlier sample* **do**
    $\mathbf{S}_{a,b} \leftarrow$ SamplePairs($\mathbf{F}_{a,b}$)
    **if** *not* AllScalarPartsEqual($\mathbf{S}_{a,b}$) **then next** ;
    **if** ***RC or EC*** **then**
      $\check{\mathbf{q}}'_{a,b} \leftarrow$ ComputeHandEyeCalibration($\mathbf{S}_{a,b}$)
      $\mathbf{I}_{a,b} \leftarrow$ GetInliersBasedOnPoseError($\mathbf{F}_{a,b}$, $\check{\mathbf{q}}'_{a,b}$, $\lambda_{t,min}$, $\lambda_{r,min}$)
    **else**
      **// RS or ES**
      $\mathbf{I}_{a,b} \leftarrow$ GetInliersBasedOnScalarPartsEquality($\mathbf{S}_{a,b}, \mathbf{F}_{a,b}$)
    **end**
    **if** $|\mathbf{I}_{a,b}| <$ *required number of inliers* **then next** ;
    $\check{\mathbf{q}}'_{a,b} \leftarrow$ ComputeHandEyeCalibration($\mathbf{I}_{a,b}$)
    $(RMSE, \mathbf{I}_{a,b}) \leftarrow$ EvaluatePairs($\mathbf{P}_{a,b}, \check{\mathbf{q}}'_{a,b}$)
    **if** $RMSE < RMSE_{best}$ **then**
      $RMSE_{best} \leftarrow RMSE$
      $\check{\mathbf{q}}_{a,b} \leftarrow \check{\mathbf{q}}'_{a,b}$
    **end**
  **end**
  return ($RMSE_{best}$, $\check{\mathbf{q}}_{a,b}$)

### 3.3.2 Refinement Step

In Sections 3.2 and 3.3 we address the global extrinsic spatiotemporal hand-eye calibration problem. However, the expected accuracy is limited mostly due to the fact that the assumed pose-trajectories are estimated individually and kept fix when aligning them to find the hand-eye calibration. A joint maximum likelihood optimization of calibration and trajectory given the measurements allows higher accuracy. This optimization is hard to solve as global problem but using the results from our global approach as an initial guess a local likelihood maximization can improve the accuracy of the calibration. We perform this joint batch estimation step with a continuous-time representation for the trajectory, as in [9, 19], and overall very similar to what is described in [19]. Except for that we use Lie group valued B-splines, [22], to represent SO(3)-trajectories instead of vector space valued B-splines in an unconstrained parameter space of SO(3) [9, 19] [3]. To solve the nonlinear optimization we use an extension of Levenberg-Marquardt to Lie groups (as documented e.g. in [22]).

More specifically, we model the joint problem with one trajectory for the moving hand frame, $H$, $\mathbf{T}_{BH}(t) =: \mathbf{X}(t)$. The eye-frame, $E$, is assumed to be rigidly connected to the hand frame by the spatial calibration, $\mathbf{T}_{HE}$, as depicted in Figure 1. The

[3] Traditional B-splines in parameter space are not equivariant with respect to transformations of the world and body frame. Therefore, for a given trajectory the local expressiveness of such a representation typically depends on where the trajectory is in that segment. Furthermore, they can go through ambiguous or unstable regions of the parameter space. The Lie-group valued B-splines we use are bi-equivariant [22] and are neither locally ambiguous nor unstable.

pose measurement timestamps for $H$ and $E$ are assumed to be connected through a fixed time-offset $\Delta t$. Their errors we assume to be generated from isotropic multivariate Cauchy distributions[4] with three degrees of freedom independently for both translation (with zero mean) and rotation (with identity mean)[5] with respect to $B$ and $W$-frame respectively. Or, if the eye is only emitting relative pose estimates (as, e.g., in visual inertial odometry), the same type of error source is assumed but with respect to the pose of the last measurement event. This yields the following negative log likelihood function, $l$, which we minimize:

$$\begin{aligned} & l\left(\mathbf{T}_{WE}, \mathbf{T}_{HE}, \Delta t, \mathbf{X} \,\middle|\, (\mathbf{T}_{WEi}, t_{Ei})_{i=1}^{k}, (\mathbf{T}_{BHi}, t_{Hi})_{i=1}^{l}\right) \\ & = \sum_{i=2}^{k_E} \rho(\| d_E\left(\mathbf{T}_{WEi-1}, \mathbf{T}_{WEi}, \mathbf{T}_{WE}(t_{Ei-1}), \mathbf{T}_{WE}(t_{Ei})\right) \|^2_{\Sigma_E}) \\ & + \sum_{i=2}^{k_H} \rho(\| d_H\left(\mathbf{T}_{BHi-1}, \mathbf{T}_{BHi}, \mathbf{T}_{BH}(t_{Hi-1}), \mathbf{T}_{BH}(t_{Hi})\right) \|^2_{\Sigma_H}), \end{aligned} \tag{2}$$

where $\mathbf{T}_{WE}(t) := \mathbf{T}_{WB}\mathbf{T}_{BH}(t-\Delta t)\mathbf{T}_{HE}$, $\mathbf{T}_{BH}(t) = \mathbf{X}(t)$, $\rho(s) = log(1+s)$ the Cauchy-loss, and $d_E$ and $d_H$ are either relative, $(\mathbf{A}', \mathbf{A}, \mathbf{B}', \mathbf{B}) \mapsto d(\mathbf{A}'^{-1}\mathbf{A}, \mathbf{B}'^{-1}\mathbf{B})$, or absolute $(\mathbf{A}', \mathbf{A}, \mathbf{B}', \mathbf{B}) \mapsto d(\mathbf{A}, \mathbf{B})$ [6]. As displacement vector $d(\mathbf{A}, \mathbf{B}) \in \mathbb{R}^6$ on SE(3) we use coordinates of $(\log_{SO(3)}(\mathbf{R}), \mathbf{u})$ with respect to a fixed positive orthonormal basis, where $\mathbf{u}$ is a translation and $\mathbf{R}$ a proper rotation such that (uniquely) $\mathbf{u} \circ \mathbf{R} := \mathbf{A}^{-1}\mathbf{B}$. Please note that $l$ becomes independent of $\mathbf{T}_{WE}$ iff $d_E$ is relative because then it cancels out in $\mathbf{B}'^{-1}\mathbf{B}$.

## 4 Datasets

In the scope of this work, we evaluated our calibration framework on three different systems. Firstly, an RGB-D sensor with a visual target in an external tracking system. Secondly, a robot arm with an RGB-D sensor mounted close to the end effector. And, lastly, we have mounted three Tango tablets on a rigid profile, and used it for recording two datasets with different motions.

**RGB-D-Sensor in external motion caption system:** In the first experiments, we recorded color images from a PrimeSense RGB-D sensor, which was tracked using a Vicon tracking system. We placed a visual target in front of the camera to be able to use the camera pose estimation described in Section 3.1.

**Robot Arm with RGB-D-Sensor:** We recorded two datasets with a UR-10 robot arm equipped with a RealSense SR300 RGB-D sensor, mounted rigidly to a sensor

[4] This is equivalent to least squares with a Cauchy loss function.

[5] Approximated with zero-mean Cauchy distributions in the Lie algebra projected to SO(3) using the exponential map.

[6] For absolute $d_E, d_H$ the corresponding first measurements, $i = 1$, are assumed to be dummy variables while the real measurements start with $i = 2$.

mount close to the end effector. The first dataset is simulated and recorded in the Gazebo robotic simulator [15]. In this dataset, we can extract the ground truth hand-eye transformation from the setup of the robot model. The second dataset is a similar setup, but recorded on a real robot. In both, the simulation and the real world experiment, there is an *AprilTag* target placed on the robot base to estimate the camera motion.

**Rig with Three Tango Tablets:** The next datasets contain pose estimations from three Google Tango tablets [10] that are rigidly mounted on an aluminum profile. The datasets are recorded in an indoor environment. In order to improve the accuracy of the Tango pose estimation, we used the Tango framework to find loop closures and create optimized localization maps based on the individual trajectories and then exported the self-localized pose estimates of the Tango tablet.

## 5 Results

Evaluating hand-eye calibrations is inherently difficult, as ground truth values are not available for real systems. In order to evaluate the initial hand-eye calibration results, we use the same evaluation method also employed for the sample evaluation in the RANSAC algorithm, i.e., we transform the pose pairs into a common frame and compute the RMSE of the position and orientation. For the datasets with more than one sensor pair, we further evaluate the accumulation of position / orientation error that occurs when all sensor pair calibrations are combined to form a loop, hence ideally resulting in the identity transform. In order to evaluate the different components of the proposed system we compare the *PF_RC*, *NF_RC*, *PF_RS*, *NF_RS*, *PF_B*, *NF_B*, *PF_EC*, *PF_ES* variants (see Section 3.3) on the datasets described in Section 4. As a refinement step, we apply the refinement described in Section 3.3.2 to the individual initial guesses.

Furthermore, we compare the runtimes of the different algorithms. All non-deterministic algorithms (i.e. *RC* and *RS*) are run 20 times and the results are accumulated using box-plots. For the Tango datasets, we additionally accumulated the measurements of all 3 hand-eye calibration pairs.

### *5.1 Time Alignment*

In order to demonstrate the importance of filtering the angular velocity norm prior to the correlation used for time-alignment, we show how the RMSE results of the *ES* algorithm improve, see Table 1 and Figure 3. For the *PrimeSense* and the *robot arm* dataset we see an improvement of the calibration result. This corresponds with the observation, that there is more noise on the orientation for those datasets. If the time offset, which is a multiple of the discrete time steps of the timestamped poses,

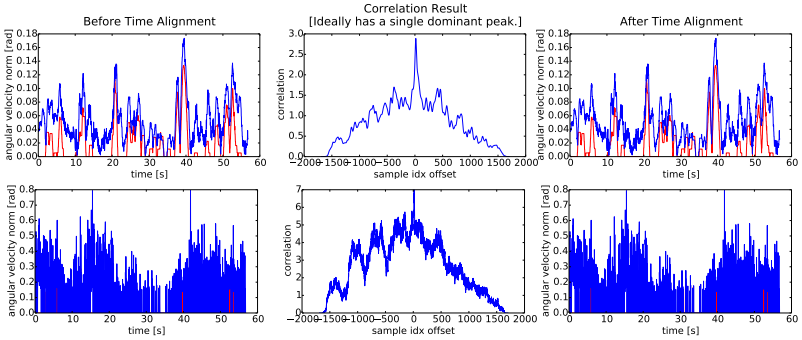


Fig. 3: Time-alignment with *(top)* and without *(bottom)* capping and low-pass filtering.

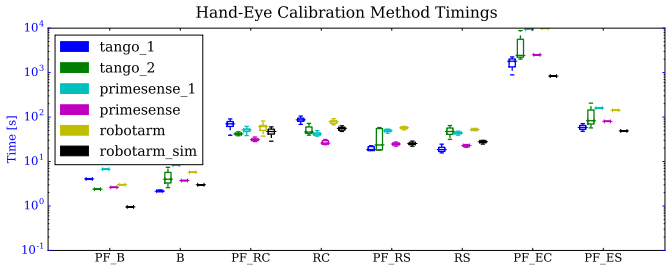


Fig. 4: The timings of the differently filtered algorithms on all datasets.

is the same with and without filtering, the results are identical, as observed for the datasets: *robot arm sim* and *Tango triplet*.

| RMSE (position/orientation) | PrimeSense | Tango triplets short | robot arm real | robot arm sim |
|---|---|---|---|---|
| filtering | (0.0213/0.0213) | (0.0364/0.6389) | (0.0118/0.6619) | (0.0034/0.2677) |
| no filtering | (0.0227/1.8399) | (0.0364/0.6389) | (0.0120/0.8972) | (0.0034/0.2677) |

Table 1: RMSE (position [m] / orientation [deg]) results for the *ES* algorithm with and without angular velocity norm filtering.

## *5.2 Hand-Eye Calibration*

We show evaluations of the runtimes, in Figure 4, and of the RMSE of position and orientation for every dataset and algorithm in Figure 5. The two different RANSAC

based algorithms (*RS* and *RC*) both outperform the *B* in terms of calibration quality, which is to be expected as the random sampling has a higher chance of finding inliers. Furthermore, both algorithms result in a calibration quality that is very close to the *ES* and *EC* algorithms, which naturally represent the upper bound without using the refinement step. The gap in calibration quality of the RANSAC based algorithms comes at the cost of runtime, i.e. *RS* and *RC* are significantly slower than the *B* algorithm. While intended as a baseline algorithm to provide an upper bound for the calibration quality of *RS*, the *ES* algorithm proves to be efficient and therefore a valid candidate. This is due to the fact, that it only requires a single sample and, therefore, the number of combinations to explore is only the number of input poses, which has been significantly reduced by the selection of informative pose pairs. That is why it is only slightly slower than the RANSAC based algorithms. The *EC* algorithm on the other hand uses a sample size of $n$ $(n >= 3)$ and, hence, has to explore a far greater number of combinations, which is reflected in the runtime plot. Surprisingly, the prefiltering of the poses generally had a negligible effect on both calibration quality and runtime, with the exception of the above mentioned exhaustive search, which would not have been feasible without it.

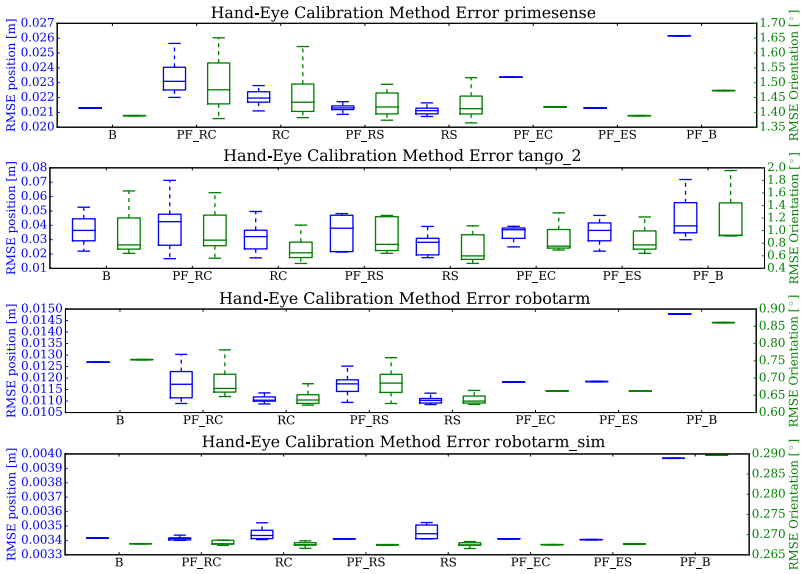


Fig. 5: Evaluation of the different filtering methods on the different datasets.

We plot the circular calibration error of the three sensors in the *Tango* datasets, see Figure 6. After the refinement step we get a mean circular position and orientation error of 4.02 mm and 0.091° for the *Tango 1* dataset, and 7.19 mm and 0.139° for the *Tango 2* dataset, which is a significant improvement over the initial calibration.

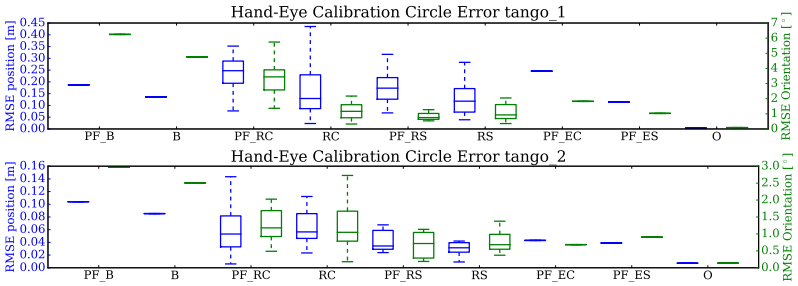


Fig. 6: The circular error of the individual methods and the combined results after applying the refinement step to the other methods, denoted with (O).

## 6 Conclusion

In this paper we presented a hand-eye calibration system that can easily be used out of the box in a variety of scenarios and environments. In order to substantiate that claim, our system is thoroughly evaluated on different datasets stemming from multiple types of platforms. Taking a holistic view on the hand-eye calibration problem, we consider aspects such as time offset estimation as well as detection and rejection of outliers. All the datasets were made publicly available together with the entire software toolbox, which was designed in a modular way to ensure extendability.

## Acknowledgement

This work was partially supported by the Swiss National Science Foundation (SNF), within the National Centre of Competence in Research on Digital Fabrication, by Google in the context of the Tango project, and by the European Union's Seventh Framework Programme for research, technological development and demonstration under the EUROPA2 project No. FP7-610603.

## References

1. M. K. Ackerman, A. Cheng, B. Shiffman, E. Boctor, and G. Chirikjian. Sensor Calibration with Unknown Correspondence: Solving AX = XB Using Euclidean-Group Invariants. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1308–1313, 2013.
2. N. Andreff, R. Horaud, and B. Espiau. Robot Hand-Eye Calibration Using Structure-from-Motion. *International Journal of Robotics Research*, 20(3):228–248, 2001.
3. J. C. K. Chou and M. Kamel. Finding the position and orientation of a sensor on a robot manipulator using quaternions. *International Journal of Robotics Research*, 10(3):240–254, 1991.

4. K. Daniilidis. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18(3):286–298, 1999.
5. A. English, P. Ross, D. Ball, B. Upcroft, and P. Corke. Triggersync: A time synchronisation tool. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6220–6226. IEEE, 2015.
6. I. Fassi and G. Legnani. Hand to Sensor Calibration: A Geometrical Interpretation of the Matrix Equation AX = XB. *Journal on Robotics Systems*, 22(9):497–506, 2005.
7. P. Furgale, T. D. Barfoot, and G. Sibley. Continuous-time batch estimation using temporal basis functions. In *2012 IEEE International Conference on Robotics and Automation*, pages 2088–2095, May 2012.
8. P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286, Nov 2013.
9. P. Furgale, C. H. Tong, T. D. Barfoot, and G. Sibley. Continuous-time batch trajectory estimation using temporal basis functions. *The International Journal of Robotics Research*, 34(14):1688–1710, 2015.
10. Google. ATAP Project Tango Google, Feb. 2014.
11. A. Harrison and P. Newman. Ticsync: Knowing when things happened. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 356–363. IEEE, 2011.
12. J. Heller, M. Havlena, A. Sugimoto, and T. Pajdla. Structure-from-Motion Based Hand-Eye Calibration Using L Minimization. In *2011 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3497–3503, 2011.
13. R. Horaud and F. Dornaika. Hand-eye Calibration. *International Journal of Robotics Research*, 14(3):195–210, 1995.
14. J. Kelly and G. S. Sukhatme. A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors. In *Experimental Robotics*, pages 195–209. Springer, 2014.
15. N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, Sept 2004.
16. J. Maye, P. Furgale, and R. Siegwart. Self-supervised calibration for robotic systems. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 473–480, June 2013.
17. E. Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, May 2011.
18. F. C. Park and B. J. Martin. Robot Sensor Calibration: Solving AX = XB on the Euclidean Group. *IEEE Transactions on Robotics and Automation*, 10(5):717–721, 1994.
19. J. Rehder, R. Siegwart, and P. Furgale. A general approach to spatiotemporal calibration in multisensor systems. *IEEE Transactions on Robotics*, 32(2):383–398, 2016.
20. J. Schmidt, F. Vogt, and H. Niemann. Robust Hand Eye Calibration of an Endoscopic Surgery Robot Using Dual Quaternions. In *Vision, Modeling, and Visualization*, pages 21–28, 2004.
21. Y. C. Shiu and S. Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB. *IEEE Transactions on Robotics and Automation*, 5(1):16–29, 1989.
22. H. Sommer, J. R. Forbes, R. Siegwart, and P. Furgale. Continuous-time estimation of attitude using b-splines on lie groups. *Journal of Guidance, Control, and Dynamics*, 2015.
23. K. H. Strobl and G. Hirzinger. Optimal hand-eye calibration. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4647–4653. IEEE, 2006.
24. Z. Taylor and J. Nieto. Motion-Based Calibration of Multimodal Sensor Extrinsics and Timing Offset Estimation. *IEEE Transactions on Robotics*, 32(5):1215–1229, 2016.
25. R. Y. Tsai and R. K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, 1989.
26. C.-C. Wang. Extrinsic Calibration of a Vision Sensor Mounted on a Robot. *IEEE Transactions on Robotics and Automation*, 8(2):161–175, 1992.