

Recursive Bayesian Calibration of Depth Sensors with Non-Overlapping Views

Florian Faion, Patrick Ruoff, Antonio Zea, and Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory (ISAS),

Institute for Anthropomatics,

Karlsruhe Institute of Technology (KIT), Germany.

florian.faion@kit.edu, patrick.ruoff@kit.edu, antonio.zea@student.kit.edu, uwe.hanebeck@ieee.org

Abstract—In this paper, we present a recursive Bayesian method to calibrate rigidly linked depth sensors with non-overlapping fields of view. The extrinsic parameters of this setup are obtained by rotating and translating both cameras, estimating the local transformations using point feature correspondences, and finally using these values to recursively find a solution to the matrix equation $A_k X = X B_k$. The algorithm is based on a Bayesian estimator, which allows the consideration of camera-specific measurement noise and permits the system to adapt naturally to changes in the extrinsic parameters. Special care was taken to keep the system free from singularities. This paper also includes a thorough evaluation based on synthetic and real data to show the effectiveness of the algorithm.

I. INTRODUCTION

Setting up a network of multiple depth sensors introduces several issues that need to be addressed. For example, the popular Microsoft KinectTM device acquires depth information using an active measurement system which projects an infrared pattern. This has important consequences in setups with overlapping fields of view, where the interference caused by overlaying patterns becomes an important factor for measurement quality. One solution for this problem is a time-multiplexing approach where measurements are taken only by a single sensor at any given time. However, this approach becomes ineffective for applications such as tracking, where it may be necessary to switch between sensors very quickly, requiring the projectors to be rapidly powered on and off. This can lead to a different set of problems, especially for Kinect devices, since existing drivers were not designed for this task.

To avoid these issues it becomes necessary to explore setups with non-overlapping fields of view. In this case, however, the sensors cannot be calibrated using standard methods where shared features are required to be visible in both cameras. This is the same challenge faced in other classes of setups, e.g., the extrinsic hand to eye calibration for robots, which has been extensively researched. The most common approach, used also in this paper, is to reduce the problem to solving the equation $A X = X B$, where A , B , and X are rigid transformation matrices.

A. Contribution

The main contribution of this paper is the development of a recursive Bayesian state estimator to obtain the relative

extrinsic calibration between two depth sensors with non-overlapping fields of view, based on local point features (Figure 1). This includes a sound stochastic modeling, as well as an experimental evaluation based on synthetic and real data.

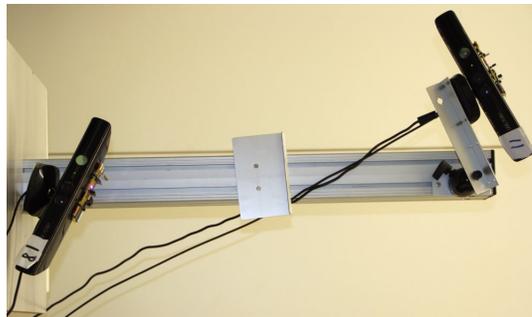


Fig. 1: Calibration setup with two rigidly linked depth sensors.

B. Related Work

In this section, we briefly refer to related methods for solving $A X = X B$. In the context of robotic hand-to-eye calibration, several non-recursive methods based on homogeneous matrices [1], quaternions [2], and Lie theory [3] have been developed. Calibration of non-overlapping standard RGB camera setups has been considered [4], [5] using non-recursive approaches, which estimate rotation and translation independent of each other. A more general solution for arbitrarily complex matrices [6] has also been explored. This paper uses a method to determine a rigid transformation from $n \geq 3$ corresponding 3D points [7]. Rotation vectors were found to be suitable for recursive pose estimation [8]. Similar to this work, an Extended Kalman Filter was applied for stereo matching [9], [10].

C. Overview

The remainder of this paper is structured as follows.

In Section II, the calibration problem is described in more detail. The motivation behind the stochastic approach is explained in Section III, together with the theoretical principles. Section IV presents a description of the recursive algorithm. The derived method is extensively evaluated in Section V using synthetic and real data experiments. Finally, Section VI

concludes this paper with a summary of the authors' insights and an outlook on future research.

II. PROBLEM FORMULATION

We consider the problem of recursively estimating the transformation \mathbf{X} between two rigidly linked depth sensors, in the following denoted as Cam_A and Cam_B , mounted on a mobile base. In each time step k , these cameras are rotated and translated while remaining rigidly connected. The exact transformation can be arbitrary, and the algorithm does not require this transformation to be known in advance.

For each time step k , both cameras produce a noisy depth image, from which a set of distinct point features is extracted. These features can be matched between successive time steps k and $k+1$, yielding a set of noisy point correspondences. For Cam_A , the measured feature i at time k is denoted by $\underline{P}_{A_i,k}$ and the corresponding feature at the following time step $k+1$ by $\underline{P}_{A_i,k+1}$. Measurements for Cam_B are denoted in a similar way as $\underline{P}_{B_i,k}$ and $\underline{P}_{B_i,k+1}$. For clarity, the feature correspondences between time steps k and $k+1$ are aggregated in the matrices \mathbf{P}_A and \mathbf{P}_B .

Applying a transformation \mathbf{H} to the linked cameras affects the local reference frames of Cam_A and Cam_B . Let \mathbf{A} and \mathbf{B} denote these local reference changes. Note that time indices are dropped for clarity. Then, assuming that the point features are corrupted with additive Gaussian measurement noise, the relationship between the measurements can be described as

$$\underline{P}_{A_i,k+1} - \underline{\mathbf{v}}_{i,k+1} = \mathbf{A} \cdot (\underline{P}_{A_i,k} - \underline{\mathbf{v}}_{i,k}),$$

and

$$\underline{P}_{B_i,k+1} - \underline{\mathbf{w}}_{i,k+1} = \mathbf{B} \cdot (\underline{P}_{B_i,k} - \underline{\mathbf{w}}_{i,k}),$$

where $\underline{\mathbf{v}}_{i,k}$, $\underline{\mathbf{w}}_{i,k}$, $\underline{\mathbf{v}}_{i,k+1}$, $\underline{\mathbf{w}}_{i,k+1}$ are assumed to be zero-mean Gaussian noise terms.

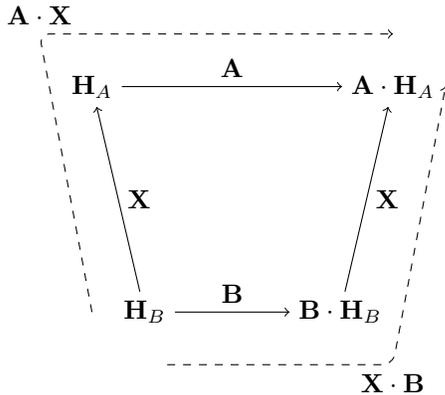


Fig. 2: Problem formulation as a commutative diagram.

Figure 2 describes the camera setup and its transformation. Let \mathbf{H}_A and \mathbf{H}_B be the original extrinsic parameters of Cam_A and Cam_B respectively. After moving the cameras, their new extrinsic parameters will be $\mathbf{A} \cdot \mathbf{H}_A$ and $\mathbf{B} \cdot \mathbf{H}_B$. Since all

involved matrices are non-singular, the extrinsic conversion between \mathbf{H}_B and $\mathbf{A} \cdot \mathbf{H}_A$ must be unique. This leads to

$$\mathbf{A}\mathbf{X} = \mathbf{X}\mathbf{B}. \quad (1)$$

The affine transformation matrix \mathbf{X} can be decomposed into a rotation matrix \mathbf{R}_X and a translation vector \underline{t}_X , represented by

$$\mathbf{X} = \begin{pmatrix} \mathbf{R}_X & \underline{t}_X \\ \underline{\mathbf{0}}^T & 1 \end{pmatrix}.$$

\mathbf{A} and \mathbf{B} can also be decomposed in a similar way

$$\mathbf{A} = \begin{pmatrix} \mathbf{R}_A & \underline{t}_A \\ \underline{\mathbf{0}}^T & 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \mathbf{R}_B & \underline{t}_B \\ \underline{\mathbf{0}}^T & 1 \end{pmatrix}.$$

Plugging \mathbf{A} , \mathbf{B} , and \mathbf{X} into Equation (1) results in

$$\begin{pmatrix} \mathbf{R}_A & \underline{t}_A \\ \underline{\mathbf{0}}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R}_X & \underline{t}_X \\ \underline{\mathbf{0}}^T & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_X & \underline{t}_X \\ \underline{\mathbf{0}}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R}_B & \underline{t}_B \\ \underline{\mathbf{0}}^T & 1 \end{pmatrix}.$$

This leads to

$$\begin{pmatrix} \mathbf{R}_A \mathbf{R}_X & \mathbf{R}_A \underline{t}_X + \underline{t}_A \\ \underline{\mathbf{0}}^T & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_X \mathbf{R}_B & \mathbf{R}_X \underline{t}_B + \underline{t}_X \\ \underline{\mathbf{0}}^T & 1 \end{pmatrix}. \quad (2)$$

Based on Equation (2), we obtain two equations, one characterizing the rotation (3×3)

$$\mathbf{R}_A \mathbf{R}_X = \mathbf{R}_X \mathbf{R}_B, \quad (3)$$

and one for translation (3×1)

$$\mathbf{R}_A \underline{t}_X + \underline{t}_A = \mathbf{R}_X \underline{t}_B + \underline{t}_X. \quad (4)$$

For any given pair of affine matrices \mathbf{A} , \mathbf{B} , Equation (3) and (4) are underdetermined. Therefore, a solution requires a minimum of two transformation steps $\{\mathbf{A}_i, \mathbf{B}_i \mid i \geq 2\}$.

III. THEORETICAL BACKGROUND

The main idea is to model the desired transformation \mathbf{X} as a random variable and apply a recursive Bayesian state estimator to solve Equation (1). While moving the cameras, pairs of transformations \mathbf{A}_k and \mathbf{B}_k , also modeled as random variables, can be extracted in each time step and then be given to the estimator as measurements. The following section describes the stochastic motivation.

It should be noted that the Bayesian estimator framework allows for dynamically updating the transformation, i.e., a model of \mathbf{X}_k where the sensors are movable in relation to each other in time. For simplicity, this paper will focus exclusively on rigidly linked cameras (constant \mathbf{X}).

A. Stochastic Modeling

In order to systematically consider uncertainties in the correspondence observations and the transformations derived from these, we model all occurring variables as random variables, assuming the conditional independence structure depicted in the following Bayesian network (Figure 3).

In this diagram, \mathbf{P}_A and \mathbf{P}_B denote the local point correspondences between given time steps k and $k+1$. The extrinsic transformation matrix \mathbf{X} is assumed to be conditionally

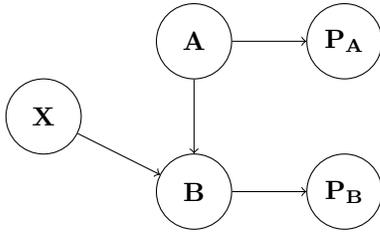


Fig. 3: Bayesian network, representing the assumed conditional independence structure of the occurring random variables.

independent from the point correspondences that determine the local transformations. Furthermore, the value of \mathbf{B} can be calculated given the values of \mathbf{X} and \mathbf{A} , since

$$\mathbf{B} = \mathbf{X}^{-1} \mathbf{A} \mathbf{X} .$$

This conditional independence structure then translates to the following factorization of the joint probability distribution

$$p(\mathbf{X}, \mathbf{A}, \mathbf{B}, \mathbf{P}_A, \mathbf{P}_B) = p(\mathbf{P}_A|\mathbf{A}) p(\mathbf{P}_B|\mathbf{B}) p(\mathbf{B}|\mathbf{A}, \mathbf{X}) p(\mathbf{A}) p(\mathbf{X}) . \quad (5)$$

According to the problem formulation, we are given the point correspondences \mathbf{P}_A and \mathbf{P}_B , from which we derive the conditional probability densities $p(\mathbf{A}|\mathbf{P}_A) = p_A(\mathbf{A})$ and $p(\mathbf{B}|\mathbf{P}_B) = p_B(\mathbf{B})$. This is performed by calculating a Monte-Carlo estimate of the densities' mean and covariance, implicitly assuming that all occurring densities are Gaussian (see Section III-B for the motivation behind this assumption). Given a prior estimate of \mathbf{X} in the form of a probability density $p(\mathbf{X})$, the posterior density can then theoretically be calculated via

$$\begin{aligned} p(\mathbf{X}|\mathbf{P}_A, \mathbf{P}_B) &= \int \int p(\mathbf{X}, \mathbf{A}, \mathbf{B}|\mathbf{P}_A, \mathbf{P}_B) d\mathbf{A} d\mathbf{B} \\ &= \int \int \frac{p(\mathbf{X}, \mathbf{A}, \mathbf{B}, \mathbf{P}_A, \mathbf{P}_B)}{p(\mathbf{P}_A, \mathbf{P}_B)} d\mathbf{B} d\mathbf{A} \\ &\stackrel{(5)}{=} \int \int \frac{p(\mathbf{P}_A|\mathbf{A}) p(\mathbf{P}_B|\mathbf{B})}{p(\mathbf{P}_A, \mathbf{P}_B)} p(\mathbf{B}|\mathbf{A}, \mathbf{X}) p(\mathbf{A}) p(\mathbf{X}) d\mathbf{B} d\mathbf{A} \\ &= \int \int \frac{p(\mathbf{A}, \mathbf{B}|\mathbf{P}_A, \mathbf{P}_B)}{p(\mathbf{B})} p(\mathbf{B}|\mathbf{A}, \mathbf{X}) p(\mathbf{X}) d\mathbf{B} d\mathbf{A} \\ &= \int \int \frac{p(\mathbf{A}|\mathbf{P}_A) p(\mathbf{B}|\mathbf{P}_B)}{p(\mathbf{B})} \delta(\mathbf{B} - \mathbf{X}^{-1} \mathbf{A} \mathbf{X}) d\mathbf{B} d\mathbf{A} p(\mathbf{X}) \end{aligned} \quad (6)$$

$$\propto \int p_A(\mathbf{A}) p_B(\mathbf{X}^{-1} \mathbf{A} \mathbf{X}) d\mathbf{A} p(\mathbf{X}) \quad (7)$$

$$= \int p_A(\mathbf{X} \mathbf{A}) p_B(\mathbf{A} \mathbf{X}) d\mathbf{A} p(\mathbf{X}) . \quad (8)$$

We have used the conditional independence structure of \mathbf{A} , \mathbf{B} and $\mathbf{P}_A, \mathbf{P}_B$ in (6) and assumed a uniform / uninformative prior over \mathbf{B} in (7). In (8), we shifted the integrand by \mathbf{X} , in order to bring the resulting likelihood to a symmetric

form. Note that this implies that we assume the underlying integration measure to be shift-invariant, which is the case for the product of the rotation group's Haar-measure with the translation group's Lebesgue-measure.

Unfortunately, the derived likelihood

$$l(\mathbf{X}) = \int p_A(\mathbf{X} \mathbf{A}) p_B(\mathbf{A} \mathbf{X}) d\mathbf{A}$$

is analytically intractable. Because of this, an Unscented Kalman Filter [11] is used to recursively estimate the calibration, which implicitly assumes that the probability distribution of \mathbf{X} , \mathbf{A} , \mathbf{B} , and in consequence the measurement vector $\mathbf{A} \mathbf{X} - \mathbf{X} \mathbf{B}$, is Gaussian. The following section elaborates further upon this assumption.

B. Representation of Rigid Transformations

Section III-A explained the difficulty of an analytical treatment of the likelihood, and the motivation for using an Unscented Kalman Filter to estimate the posterior density of \mathbf{X} . As mentioned above, this assumes that the distribution of the transformations is Gaussian. Since the Gaussian distribution is defined in coordinate space, the resulting distribution depends on the chosen parametrization of the Euclidean group $SE(3)$, in particular the rotation part. This raises the question of which transformation representation is the most appropriate.

We tested three popular representations, i.e., rotation vectors (equivalent to axis-angle representation), quaternions, and euler angles, with Royston's H-Test [12] to rate the quality of the Gauss assumption for individual transformation representations. To this end, 1000 rigid transformations were randomly generated. For each transformation, $100 \times 100 = 10000$ noisy point correspondences were sampled. From these, 100 least square estimates [7] were calculated, each based on 100 correspondences. These estimates were interpreted as i.i.d. samples of the posterior transformation's distribution and tested for Gaussianity with Royston's H-Test.

Evaluation of the Representations: Table I shows the empirical rejection rate over 1000 trials for a significance level of 5% depending on the chosen rotation representation and the standard deviation of the point features. As one can see, for low standard deviations the empirical rejection rate is only slightly increased compared to the p-value, which is consistent with the hypothesis that the posterior distribution is nearly Gaussian. For higher standard deviations, the null-hypothesis of a Gaussian distribution in coordinate space gets rejected more often, as the true geometry of $SO(3)$ can no longer be neglected. Furthermore, we can see that the rotation vector representation has the lowest rejection rate. This can be explained by noting that quaternions are not a minimal representation of rotations, i.e., not a chart on $SO(3)$. Instead, all valid quaternions lie on the unit-sphere, which naturally makes a Gaussian assumption problematic. The Euler angle representation, on the other hand, possesses singularities like Gimbal locks, around which the natural metric is heavily distorted.

Because of this analysis, in the following all transformations will be described using the rotation vector form.

Variance	10^{-3}	10^{-2}	10^{-1}
Rotation Vector	6.7%	6.3%	7.1%
Quaternion	7.7%	7.5%	8.8%
Euler Angles	8.5%	8.8%	13.6%

TABLE I: Rejection rate of Royston’s H-Test with $p = 0.05$ for different representations of transformation.

Input: $\underline{\mathbf{x}}_k^e \sim \mathcal{N}(\hat{\underline{\mathbf{x}}}_k, \mathbf{C}_{\underline{\mathbf{x}}_k^e})$, point correspondences $\mathbf{P}_A, \mathbf{P}_B$
1: $\underline{\mathbf{x}}_{k+1}^p = \text{predict } \underline{\mathbf{x}}_k^e$
2: $\underline{\mathbf{y}}_{k+1} = \text{compute } \begin{pmatrix} \underline{\mathbf{u}}_{A_{k+1}} \\ \underline{\mathbf{t}}_{A_{k+1}} \end{pmatrix}$ and $\begin{pmatrix} \underline{\mathbf{u}}_{B_{k+1}} \\ \underline{\mathbf{t}}_{B_{k+1}} \end{pmatrix}$ from $\mathbf{P}_A, \mathbf{P}_B$
with $\underline{\mathbf{y}}_{k+1} \sim \mathcal{N}(\hat{\underline{\mathbf{y}}}_{k+1}, \mathbf{C}_{\underline{\mathbf{y}}_{k+1}})$
3: $\underline{\mathbf{x}}_{k+1}^e = \text{update } \underline{\mathbf{x}}_{k+1}^p$ with $\underline{\mathbf{y}}_{k+1}$
Output: $\underline{\mathbf{x}}_{k+1}^e$

Fig. 4: Recursive calibration step.

IV. RECURSIVE ALGORITHM

This section explains in detail the recursive algorithm to solve Equation (1).

A. Bayesian Estimator

A Bayesian state estimator can be used to recursively estimate the state parameters $\underline{\mathbf{x}}_k$ ¹ modeled as random vectors. Bayesian state estimation by design includes a time update, which predicts the estimated state to a future time step $k + 1$, and a measurement update, which corrects a predicted state using a sensor measurement $\underline{\mathbf{y}}_{k+1}$.

B. State Representation

The desired rigid transformation from Cam_A to Cam_B can be defined as

$$\underline{\mathbf{x}} = \begin{pmatrix} \underline{\mathbf{u}}_X \\ \underline{\mathbf{t}}_X \end{pmatrix},$$

where $\underline{\mathbf{u}}_X$ is a rotation vector and $\underline{\mathbf{t}}_X$ is a translation vector. As mentioned above, $\underline{\mathbf{x}}$ can be modeled as a random vector with $\underline{\mathbf{x}}_k^e \sim \mathcal{N}(\hat{\underline{\mathbf{x}}}_k, \mathbf{C}_{\underline{\mathbf{x}}_k^e})$. In Section II, point correspondences were defined as associated local point features from one time step k to the following time step $k + 1$. This assumes an underlying tracking mechanism which provides local point correspondences \mathbf{P}_A , and \mathbf{P}_B for each time step.

Given a state estimate $\underline{\mathbf{x}}_k^e$ and point correspondences $\mathbf{P}_A, \mathbf{P}_B$, the algorithm from Figure 4 can be applied for determining $\underline{\mathbf{x}}_{k+1}^e$.

C. Time Update

First (Line 1), the estimated state $\underline{\mathbf{x}}_k^e$ is predicted to the time step $k + 1$, assuming it evolves according to a certain system

¹In this paper we denote the predicted states by $\underline{\mathbf{x}}_k^p$ and the estimated states (after measurement updates) by $\underline{\mathbf{x}}_k^e$.

model. As we focus on rigidly mounted cameras, we consider a random walk behavior with

$$\underline{\mathbf{x}}_{k+1} = \underline{\mathbf{x}}_k + \underline{\mathbf{v}},$$

where $\underline{\mathbf{v}} \sim \mathcal{N}(\underline{\mathbf{0}}, \mathbf{C}_x)$ represents the montage uncertainty.

D. Measurement Update

From the point correspondences $\mathbf{P}_A, \mathbf{P}_B$ between time step k and $k + 1$, n least square estimates of the local transformations $\{\underline{\mathbf{y}}_{k+1}^i \mid i = 1, \dots, n\}$ with

$$\underline{\mathbf{y}}_{k+1}^i = \begin{pmatrix} \underline{\mathbf{u}}_{A_{k+1}}^i \\ \underline{\mathbf{t}}_{A_{k+1}}^i \\ \underline{\mathbf{u}}_{B_{k+1}}^i \\ \underline{\mathbf{t}}_{B_{k+1}}^i \end{pmatrix}$$

are calculated [7]. These transformations are assumed to be drawn from an underlying random vector

$$\underline{\mathbf{y}}_{k+1} = \begin{pmatrix} \underline{\mathbf{u}}_{A_{k+1}} \\ \underline{\mathbf{t}}_{A_{k+1}} \\ \underline{\mathbf{u}}_{B_{k+1}} \\ \underline{\mathbf{t}}_{B_{k+1}} \end{pmatrix},$$

defined by $\underline{\mathbf{y}}_{k+1} \sim \mathcal{N}(\hat{\underline{\mathbf{y}}}_{k+1}, \mathbf{C}_{\underline{\mathbf{y}}_{k+1}})$. As an approximation, it is assumed that

$$\hat{\underline{\mathbf{y}}}_{k+1} \approx \mathbb{E} \left[\{\underline{\mathbf{y}}_{k+1}^i \mid i = 1, \dots, n\} \right],$$

and

$$\mathbf{C}_{\underline{\mathbf{y}}_{k+1}} \approx \text{Cov} \left(\{\underline{\mathbf{y}}_{k+1}^i \mid i = 1, \dots, n\} \right).$$

Measurement Equation: The measurement equation can be obtained from Equation (1). Time indices are omitted in the derivation for clarity. At first, rearranging Equation (3), and (4) results in

$$\underline{\mathbf{0}} = \mathbf{R}_A \mathbf{R}_X - \mathbf{R}_X \mathbf{R}_B, \quad (9)$$

and respectively

$$\underline{\mathbf{0}} = \mathbf{R}_A \underline{\mathbf{t}}_X + \underline{\mathbf{t}}_A - \mathbf{R}_X \underline{\mathbf{t}}_B - \underline{\mathbf{t}}_X. \quad (10)$$

These equations need to be adapted to work with rotation vectors. This can be performed through the \exp and \log mapping, where \exp denotes the conversion from rotation vector to rotation matrix, and \log denotes the conversion from rotation matrix to rotation vector. These mappings are given by Rodrigues formula (see Appendix A, B) and allow Equation (9) and (10) to be rearranged into

$$\underline{\mathbf{0}} = \log(\exp(\underline{\mathbf{u}}_A) \exp(\underline{\mathbf{u}}_X)) - \log(\exp(\underline{\mathbf{u}}_X) \exp(\underline{\mathbf{u}}_B)),$$

and

$$\underline{\mathbf{0}} = \exp(\underline{\mathbf{u}}_A) \underline{\mathbf{t}}_X + \underline{\mathbf{t}}_A - \exp(\underline{\mathbf{u}}_X) \underline{\mathbf{t}}_B - \underline{\mathbf{t}}_X.$$

Since both equations are 3×1 , composing results in

$$\underline{\mathbf{0}} = \begin{pmatrix} \log(\exp(\underline{\mathbf{u}}_A) \exp(\underline{\mathbf{u}}_X)) - \log(\exp(\underline{\mathbf{u}}_X) \exp(\underline{\mathbf{u}}_B)) \\ \exp(\underline{\mathbf{u}}_A) \underline{\mathbf{t}}_X + \underline{\mathbf{t}}_A - \exp(\underline{\mathbf{u}}_X) \underline{\mathbf{t}}_B - \underline{\mathbf{t}}_X \end{pmatrix}.$$

Thus, the implicit measurement equation $h(\underline{x}, \underline{y})$ is given by

$$h(\underline{x}, \underline{y}) = \begin{pmatrix} \log(\exp(\underline{u}_A) \exp(\underline{u}_X)) - \log(\exp(\underline{u}_X) \exp(\underline{u}_B)) \\ \exp(\underline{u}_A) \underline{t}_X + \underline{t}_A - \exp(\underline{u}_X) \underline{t}_B - \underline{t}_X \end{pmatrix},$$

which generates pseudo-measurements in the form of

$$h(\underline{x}, \underline{y}) = \underline{0}.$$

V. EVALUATION

This section describes the evaluation of the recursive algorithm using a depth sensor calibration scenario. Two approaches are used. First, a synthetic scenario is considered, comparing the algorithm results while using different noise intensities. Second, the calibration of two Kinect sensors using real measurements is evaluated.

A. Synthetic Data

The ground truth was specified to be the rigid transformation $\underline{x} = [0, \frac{\pi}{3}, 0, 2, 0, 0]^T$ from Cam_A to Cam_B . The setup is visualized in Figure 5. The fields of view, colored in transparent gray, are non-overlapping. The Bayesian state estimator was initialized with $\underline{x}_0 \sim \mathcal{N}(\hat{\underline{x}}_0, \mathbf{C}_0)$, where $\hat{\underline{x}}_0 = [0, 0, 0, 0, 0, 0]^T$ and $\mathbf{C}_0 = \text{diag}(2 \cdot 10^{-4}, 2 \cdot 10^{-4}, 2 \cdot 10^{-4}, 1, 1, 1)$. In each time step k , one randomly generated rigid transformation \mathbf{H}_k was applied to both Cam_A and Cam_B . 100 corresponding point features were generated and distorted by a zero mean Gaussian noise with covariance matrices $\sigma^2 \cdot \mathbf{I}$ for both sensors. We constrained the rigid transformations \mathbf{H}_k to the x - y -plane, simulating a common indoor vehicle. In Figure 6 the root mean square error (RMSE) is depicted, averaged over 20 runs.

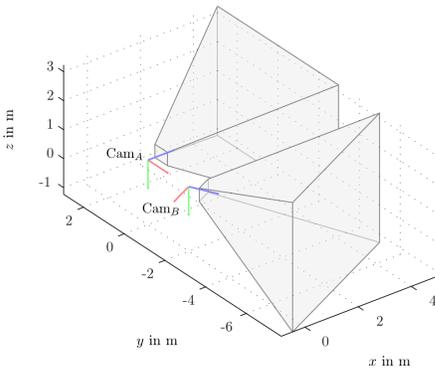


Fig. 5: Synthetic calibration setup with two rigidly linked depth sensors Cam_A and Cam_B .

B. Real Data

The real camera experiment consisted of calibrating two rigidly linked Kinect RGBD cameras (Figure 1). A chessboard with 10×7 inner corners was used to obtain features. For each time step k , the algorithm detected corners in the RGB image, then used the depth information to project the features in 3D space. Since an RGBD sensor provides the full 3D information about a point, it was not necessary to know the distances between the corners beforehand, or even for the chessboard to be flat. These k points were then compared the

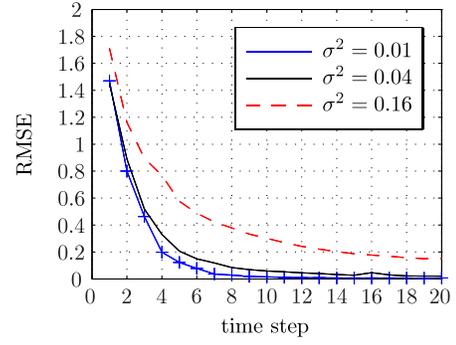


Fig. 6: RMSE of calibration steps for different σ^2 , averaged over 20 runs.

points found at $k+1$ and the transformation between them was found by minimizing the square distances. The resulting matrix was then converted to rotation vector form. The uncertainty of the transformation was calculated by using a Monte-Carlo transform with $5 \cdot 10^4$ samples.

The Bayesian state estimator was also initialized with $\underline{x}_0 \sim \mathcal{N}(\hat{\underline{x}}_0, \mathbf{C}_0)$, where $\hat{\underline{x}}_0 = [0, 0, 0, 0, 0, 0]^T$ and $\mathbf{C}_0 = \text{diag}(2 \cdot 10^{-4}, 2 \cdot 10^{-4}, 2 \cdot 10^{-4}, 1, 1, 1)$. Figure 8 shows the root mean square error (RMSE) for this experiment.

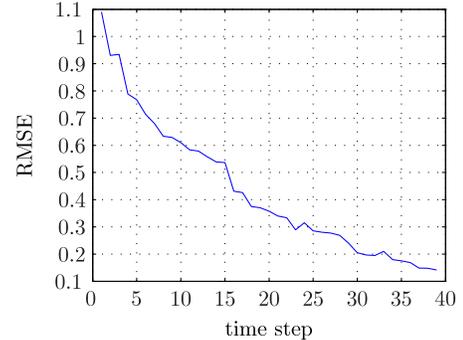


Fig. 8: RMSE of calibration steps for experiment using real data (Figure 1).

VI. CONCLUSION

In this work, a recursive algorithm for solving the calibration problem of non-overlapping depth sensors was presented. The approach is based on a Bayesian state estimator. The rigid transformation between two sensors was modeled as a random vector and recursively updated with local point correspondences of both sensors.

This novel approach considers sensor-specific uncertainties and simultaneously calibrates both the rotation and translation. Due to the recursive design, also time-dependent changes of the calibration can be modeled.

The evaluation has shown a very good convergence of the algorithm and robustness against singular movements of the base. These results were confirmed by real data experiments. It should be mentioned that the developed algorithm performs

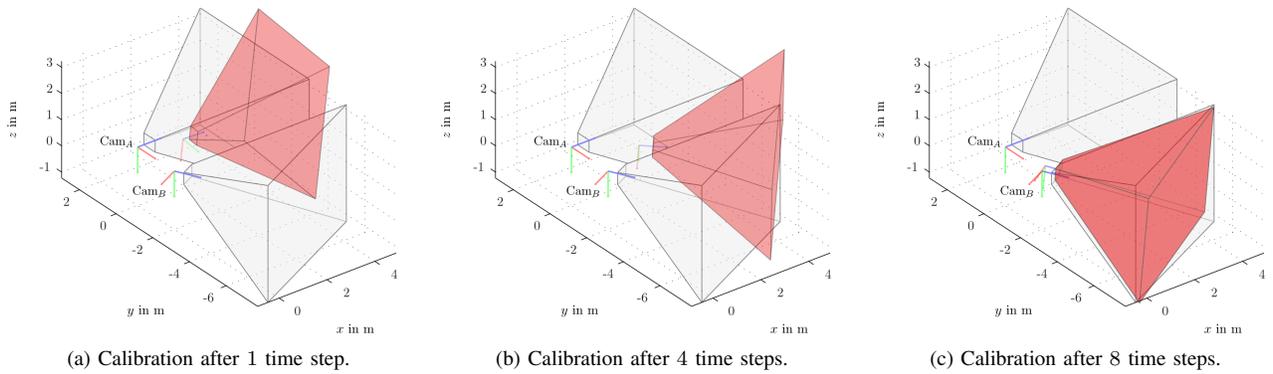


Fig. 7: Scheme of a synthetic data experiment. Light gray denotes the groundtruth, the estimation of Cam_B is colored in red.

at real time on standard hardware so that calibration can be run online.

A. Future Work

The extension of the algorithm to work with n sensors is straightforward by adapting the state and the measurement equation. A direct usage of the point correspondences as measurements also would be a desirable improvement. Given a stochastic sensor model, the uncertainty of the local transformations could be calculated analytically. Furthermore, additional sensors such as inertial measurement units (IMU) could be integrated in the calibration. Finally, combining the presented calibration approach with a scene reconstruction method like KinectFusion [13] would allow for simultaneous reconstruction with more than one Kinect.

APPENDIX

A. Mapping \exp from rotation vector \underline{u} to rotation matrix \mathbf{R}

The exponential map $\exp : so(3) \rightarrow SO(3)$ maps a tangent vector at the identity element, in our case represented by a rotation vector \underline{u} , to the corresponding rotation group element, which is represented by a rotation matrix \mathbf{R} , by

$$\begin{aligned} \exp(\underline{u}) &= e^{[\underline{u}]_{\times}} \\ &= \mathbf{I} + \sin \theta [\underline{k}]_{\times} + (1 - \cos \theta) (\underline{k} \underline{k}^{\top} - \mathbf{I}), \end{aligned}$$

where e^A denotes the matrix exponential and

$$[\underline{u}]_{\times} = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix},$$

as well as

$$(\underline{k}, \theta) = \left(\frac{\underline{u}}{|\underline{u}|}, |\underline{u}| \right).$$

The second identity is known as Rodrigues' rotation formula.

B. Mapping \log from rotation matrix \mathbf{R} to rotation vector \underline{u}

The logarithmic map $\log : SO(3) \rightarrow so(3)$ is defined as the inverse of the exponential map. Given a rotation matrix \mathbf{R} , the rotation angle can be calculated by

$$\theta(\mathbf{R}) = \arccos \left(\frac{\text{trace}(\mathbf{R}) - 1}{2} \right)$$

and the corresponding normalized rotation axis by

$$\underline{k}(\mathbf{R}) = \frac{1}{2 \sin(\theta)} \begin{pmatrix} \mathbf{R}(3,2) - \mathbf{R}(2,3) \\ \mathbf{R}(1,3) - \mathbf{R}(3,1) \\ \mathbf{R}(2,1) - \mathbf{R}(1,2) \end{pmatrix}.$$

The rotation vector is then given by

$$\log(\mathbf{R}) = \theta(\mathbf{R}) \underline{k}(\mathbf{R}).$$

REFERENCES

- [1] Y. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX=XB$," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 1, pp. 16–29, 1989.
- [2] J. C. K. Chou and M. Kamel, "Finding the Position and Orientation of a Sensor on a Robot Manipulator Using Quaternions," *The International Journal of Robotics Research*, vol. 10, no. 3, pp. 240–254, Jun. 1991.
- [3] F. Park and B. Martin, "Robot sensor calibration: solving $AX=XB$ on the Euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.
- [4] S. Esquivel, F. Woelk, and R. Koch, "Calibration of a multi-camera rig from non-overlapping views," *Pattern Recognition*, pp. 82–91, 2007.
- [5] P. Lébraly, E. Royer, O. Ait-Aider, and M. Dhome, "Calibration of Non-Overlapping Cameras—Application to Vision-Based Robotics," *Proceedings of the British Machine Vision Conference 2010*, pp. 10.1–10.12, 2010.
- [6] F. De Terán, F. M. Dopico, N. Guillery, D. Montealegre, and N. Reyes, "The solution of the equation $AX+X*B=0$," Tech. Rep., 2011.
- [7] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, p. 629, Apr. 1987.
- [8] F. Beutler, M. F. Huber, and U. D. Hanebeck, "Instantaneous Pose Estimation using Rotation Vectors," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009) in Taipei, Taiwan*, Apr. 2009, pp. 3413–3416.
- [9] M. Ragab, K. Wong, J. Chen, and M. Chang, "EKF Based Pose Estimation using Two Back-to-Back Stereo Pairs," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 6. IEEE, 2007, pp. VI–137.
- [10] J.-H. Kim and M. J. Chung, "Absolute motion and structure from stereo image sequences without stereo correspondence and analysis of degenerate cases," *Pattern Recognition*, vol. 39, no. 9, pp. 1649–1661, Sep. 2006.
- [11] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Computer Engineering*, vol. 92, no. 3, 2004.
- [12] A. Trujillo-Ortiz, R. Hernandez-Walls, K. Barba-Rojo, and L. Cupul-Magana, "Roystest: Royston's Multivariate Normality Test. A MATLAB file," 2007.
- [13] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, a. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136, Oct. 2011.