Proceedings of the 2011 IEEE
International Conference on Robotics and Biomimetics
December 7-11, 2011, Phuket, Thailand

Spatio-Temporal Initialization for IMU to Camera Registration

Elmar Mair, Michael Fleps, Michael Suppa, and Darius Burschka

Abstract— **Inertia-visual sensor systems become more and more popular in mobile robotics. They allow for global and drift-free localization at high dynamics. Cameras and inertial measurement units (IMUs) are complementary sensors which mutually enhance if correctly fused. However, this complementary nature brings major problems for the IMU to camera registration. Several solutions to compute the spatial alignment are described in literature. In this work, we want to stress the importance of temporal alignment and compare two methods for determining the temporal displacement of sensor measurements. The presented temporal registration can be used as independent preprocessing step without any knowledge about the spatial relation. Further, we present closed-form methods to initialize the angular alignment of the IMU and the camera which can also be applied to setups with gyroscopes only. If high accuracies are required this result can be used to initialize any filter or batch-optimization method to improve convergence and reduce processing time. Simulations and experiments illustrate the presented methods and underline the importance of temporal alignment.**

I. Introduction

Vision based localization and navigation have a large impact on robotic applications. There are several reasons for the success of cameras, *e.g.*, their compactness, low-power consumption and passivity. However, there is a number of drawbacks that include the large processing costs, motion blur and the restricted field of view, which effectively reduce the dynamics of the underlying system. Nevertheless, high dynamics become more and more demanding in robotic applications.

Such high dynamics are the strength of gyroscopes and accelerometers. An inertial measurement unit (IMU) consists of three gyroscopes and three accelerometers pairwise aligned to three orthogonal axes. Nowadays, especially the lightweight and cheap MEMS sensors are used to capture the angular velocity and the acceleration in 3D space. These sensors run at a high sampling rate, but they are quite noisy and suffer from biases. In several applications it has already been shown that the complementary dynamic operating bandwidths of cameras and IMUs result in a powerful combination [1], [2], [3], [4].

Several visual-inertial calibration techniques came up in the last years. They can be categorized in three different classes: closed-form solutions by reducing the system's complexity [5], [6], Kalman Filter based approaches [7], [8], [9] and methods which make use of optimization techniques [10], [11], [12], [13]. The first type of solution allows for an easy computation of the spatial alignment but requires cumbersome setups, especially for the translational alignment. Further, it is insensitive to a temporal misalignment, but any error in the physical setup influences the calibration result. Various Extended or Unscented Kalman Filter approaches exist which estimate the spatial alignment as filter state. These methods process the measurements only sequentially, which makes them online capable, but it also lessens the registration quality. Additionally, they suffer from inaccuracies due to the well-known approximations applied in the filter. The last group of solutions uses optimization techniques to determine the spatial registration of the two sensors. This turns them to offline calibration methods, but which in general achieve higher accuracies.

The filter and optimization approaches are computational expensive and sensitive to local optima. Hence, a good starting point for the spatial alignment is crucial and reduces the convergence time significantly. In the following we present a modified hand-eye calibration method which determines the spatial alignment between IMU and camera in closed-form.

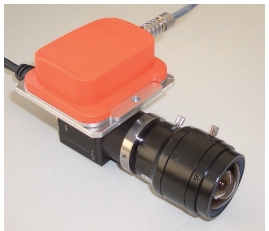

Fig. 1. A PointGrey Flea2G camera with wide angle lens and a Xsens-MTi IMU (orange box) as it is used in our experiments.

Most IMU-camera calibration approaches neglect an important issue for registration, the temporal alignment. Especially for highly dynamic motions, where the signal to noise ratio of the IMU becomes practical, an accurate alignment is crucial. If the sensors contain a clock and provide a time-stamp for their samples, one can just synchronize the clocks or the time-stamps to temporally align the measurements. Several solutions exist to synchronize clocks (an overview is given in [14]) or the time-stamps (a survey is given in [15]). However, in general cameras and IMUs provide just the sensor data and sometimes also a sample counter, but the first real time-stamp is usually only set by the driver on the host computer. Hence, a different synchronization concept must be applied.

This work is supported in part within the German Aerospace Center (DLR) and within the Technische Universität München.

E. Mair and D. Burschka are with the Department of Informatics, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany
{elmar.mair,burschka}@cs.tum.edu

M. Fleps and M. Suppa is with the Institute for Robotics and Mechatronics, German Aerospace Center (DLR), Münchner Str. 20, 82234 Wessling, Germany
{michael.fleps,michael.suppa}@dlr.de

978-1-4577-2138-0/11/$26.00 © 2011 IEEE

Various approaches exist which describe how to fuse measurements which suffer from time delays in filters. Some of them assume that the delay is known apriori [16], [17]. Li and Leung estimate the time delay in an UKF [18]. Julier and Uhlmann show how the random and unknown delays of measurements can be fused using the covariance union algorithm [19]. Tungadi and Kleeman estimate the time delay between a laser-range sensor and the odometry measurements of a mobile robot by computing the phase shift of a periodic motion [20]. They do not consider any error introduced by slippage or outlier of the range samples. Further, for their method the spatial alignment of the measurements is required.

Recently, Kelly and Sukhatme presented an approach to estimate the time delay between a proprioceptive and an exteroceptive sensor [21]. The so called Time Delay Iterative Closest Point (TD-ICP) algorithm tries to minimize the distance between the orientation trajectory in 3D space by estimating the angular and temporal alignment between the sensors. Estimating the angular trajectory by the IMU requires an error-prone strap-down computation and also the camera orientation would suffer from drift if no global landmarks, like *e.g.* a checkerboard, are provided. Further, in this work any jitter of the time-stamps is neglected.

In a previous work we described a batch-based optimization algorithm to estimate the spatial IMU camera alignment [13]. This could easily be enhanced by a further optimization parameter which denotes the time delay. However, the computational complexity would increase significantly because the B-spline matrices representing the trajectories would not be constant anymore and the convergence capability would probably be reduced due to an additional degree of freedom.

Estimating the temporal and spatial alignment simultaneously allows the delay estimate to compensate for the spatial alignment error. It seems reasonable to avoid such a compensation and assume that the temporal and the spatial alignment are not correlated. By computing the time delay in a pre-processing step of the spatial registration rejects such a correlation and prevents any influence of the time lag estimation by the spatial alignment error. In the following we present and compare two methods which compute the temporal alignment independent of the spatial one.

In the next section we describe the single steps to compute the measurement delay between the sensors. Section III describes a closed-form estimation of the angular alignment of gyroscopes and camera. Simulations and experiments are shown in Section IV.

II. Temporal Alignment

Any calibration step starts with the data acquisition. While the calibration approaches for most sensors are based on static acquisition, gyroscopes need some dynamics to provide a reasonable signal to noise ratio. Hence, the temporal alignment of the sensors becomes crucial. Most sensors do not provide any time-synchronization mechanism nor do they provide a time-stamp in their data packages. Thus, the only hint to "guess" the measurement time is the time-stamp from the driver on the host PC when the sample is received. However, this time-stamp suffers from

- **delay**, due to the acquisition time, buffers on the sensors or/and on the host interface and due to the bus used for transmission,
- **jitter**, the driver on the host side has to be scheduled in order to set the time-stamp,
- **data jams**, the processor is busy with tasks of higher priority and cannot process the arriving samples - these are buffered and all get almost the same time-stamp as soon as the driver becomes active again.

Thus, the problem arises how to interpret these time-stamps and whether it is reasonable at all to make use of them. In three steps we correct the time-stamp sequences of both sensors and align them: first, the sampling period has to be estimated, then missing samples and data jams have to be detected and fixed so that, finally, the sample sequences can be aligned.

One assumption can be made which holds for most sensors, namely, that the sensor itself acquires the measurements with a constant frequency. Calibration sequences are usually rather short, thus, temperature dependent variations can be neglected at this point. Hence, the first thing to do is to estimate the sample period. Therefore, we compute the median, $med_{\Delta t}$, of all differences between two consecutive time-stamps. This rough, but robust, estimate of the sample period is then used to detect all valid sample differences, where valid is defined as a time interval Δ_i between the time-stamps t_{i-1} and t_i which varies only up to half the sampling period. Thus, we define the indicator function $v\left(\Delta_i\right)$ as

$$v\left(\Delta_i\right) = \begin{cases} 1, & \text{if } \frac{1}{2}\, med_{\Delta t} < \Delta_i < \frac{3}{2}\, med_{\Delta t} \\ 0, & \text{else} \end{cases} . \quad (1)$$

Doing so, excessively long and short acquisition times resulting from missing samples and data jams respectively are rejected while allowing for up to $50\,\%$ jitter. The acquisition period, $\overline{m}_T$, can then be computed as the mean of all valid time differences.

Now, that we know the acquisition period, we can remove jitter, look for missing samples and sample jams and, thus, provide equidistant time-stamps. Data jams are characterized as long gaps between the time-stamps with a sequence of extremely short time differences following. A sample jam can only be recovered if the number of samples after the gap until the first valid sample interval corresponds to the number necessary to fill the gap. In case there are too few samples to fill the gap, all samples from the jam have to be rejected, because it is not possible to figure out which samples have not been buffered. This follows the strategy to reject samples rather than use false measurements. In case the sensor provides a counter, it can be used to detect missing samples and to partially recover data jams even in the absence of some measurements. The residual gaps in the time-stamp sequence represent missing samples and both, the time and measurement sequence, can be filled with values "NA" at this points, denoting unavailable samples.

The sequences have now temporally equidistant samples with period $\overline{m}_T$. We can estimate the jitter-free time-stamp of the first sample, $\overline{t}_0$, by computing the mean of all deviations

$$\overline{t}_0 \,=\, \frac{1}{|S_v|} \sum_{i \in S_v} \left(t_i \,-\, i\,\overline{m}_T \right) \tag{2}$$

with $S_v = \{ i \,|\, i \in \{1..N-1\} \,:\, v\,(t_i - t_{i-1}) = 1 \}$ denoting the set of all valid time-stamps and N being the number of samples. Finally, we can align the sequences.

In the following ${}^{C}\boldsymbol{R}_{\Delta_i}$ denotes the inter-frame direct cosine matrix (DCM) resulting from the image based motion estimation and ${}^{G}\dot{\phi}_t$, ${}^{G}\dot{\chi}_t$ and ${}^{G}\dot{\psi}_t$ represents the angular velocity measured by the gyroscopes at time instance t. In our notation the upper left letter indicates the sensor or frame of reference. Further, $\boldsymbol{p}$ will denote the Angle-Axis representation of a rotation with absolute angle θ and unit length rotation axis $\hat{\boldsymbol{p}}$ as used in [22]. To solve the ambiguity of sign of the Angle-Axis representation, the absolute angle is chosen to be always positive.

$$\boldsymbol{p} \,=\, 2\,\sin\left(\frac{\theta}{2}\right) \hat{\boldsymbol{p}} \quad | \quad \theta \geqslant 0 \tag{3}$$

While the frames in which the camera and the gyroscopes measure the rotations may be different, the measured absolute angular velocities $\dot{\theta}_t$ are frame independent and, hence, equal up to an unknown measurement error e_G and e_C.

$${}^{G}\dot{\theta}_t + e_G \,=\, {}^{C}\dot{\theta}_t + e_C \tag{4}$$

The absolute rotational velocity of the camera at the temporal midpoint between two images, $t_{i-0.5}$, can be computed by

$${}^{C}\dot{\theta}_{t_{i-0.5}} = \frac{{}^{C}\theta_{\Delta_i}}{{}^{C}\overline{m}_T} \text{ , with } \quad t_{i-0.5} = t_i - \frac{{}^{C}\overline{m}_T}{2} \tag{5}$$

In the following two different approaches to temporally align the camera and the gyros are presented.

A. Temporal Alignment by Cross-Correlation

One way to find the time delay between the sensors is to solve following problem:

$$\delta t_{G2C} = \arg\max_{\delta t} \left(\sum_{i \in S_v} {}^{G}\dot{\theta}_{t_i + \delta t} \, {}^{C}\dot{\theta}_{t_i} \right) \tag{6}$$

A brute-force solution to this problem is to use cross-correlation, which has the nice property to be robust against white noise. The conventional cross-correlation allows sequences to be aligned only up to sampling accuracy. To overcome this problem, a higher sampling resolution with sampling interval Δt can be used by interpolating the sequences. In the following we assume, without loss of generality, that the time interval of the acquired gyro measurements is longer than the one of the camera. The problem to solve becomes

$$\delta t_{G2C} = \Delta t \, \arg\max_{m} \left(\{ \text{xcorr}\,(m) \}_{m=-N_x}^{N_x} \right) \tag{7}$$

with $N_x = N_C \frac{\overline{m}_{T_C}}{\Delta t}$ and N_C denoting the total number of valid and invalid camera samples. The set $\{\cdot\}_{i=A}^{B}$ contains all elements for $i \in [A \ldots B] \subset \mathbb{Z}$. The cross-correlation function is defined as

$$\text{xcorr}\,(m) = \begin{cases} \frac{1}{N_x} \sum_{i=0}^{M} {}^{G}\dot{\theta}_{i\,\Delta t + m\,\Delta t} \, {}^{C}\dot{\theta}_{i\,\Delta t} & \text{if} \quad m \geqslant 0 \\ \frac{1}{N_x} \sum_{i=0}^{M} {}^{G}\dot{\theta}_{i\,\Delta t} \, {}^{C}\dot{\theta}_{i\,\Delta t + m\,\Delta t} & \text{else} \end{cases} \tag{8}$$

whereas $M = \frac{\overline{m}_{T_C}(N_C - 1) - m\,\Delta t}{\Delta t}$ which clips the overlapping sequences. The interval for m can also be reduced to speed up processing. The interpolation causes an error which is proportional to the kind of the chosen approximation. Further, if the temporal displacement is too large, cross-correlation may not find the optimal fit anymore because the overlap of the sequences becomes too small.

B. Temporal Alignment by Phase Congruency

Another way to address this problem is to evaluate the measurement sequence in frequency domain. Therefore, the measurements have to be transformed in the frequency domain resulting in $\mathcal{F}\,(\omega)$. The amplitude and the phases of the signal can be computed by

$$A\,(\omega) \,=\, |\mathcal{F}\,(\omega)\,| \quad , \quad \varphi\,(\omega) \,=\, \arg\left(\mathcal{F}\,(\omega)\right) \tag{9}$$

The phase shift between the common frequencies reveals the temporal alignment, ${}^{G}\delta t_C$. Considering the amplitude, high frequencies consist mainly of the measurement noise and the lower frequencies contain the bias of the gyroscopes. Furthermore, frequency bin 0 has a large spectral leakage because the absolute angles are defined to be only positive. Therefore, we ignore all frequencies before the first minimum in the spectrum. To suppress outliers and noise we introduce the following normalized weighting function which amplifies similar measurements with large amplitude

$$w\,(\omega) = \left(\sum_{\omega \in \mathcal{F}_v} \frac{1}{w'\,(\omega)} \right) w'\,(\omega) \quad \text{with} \tag{10}$$

$$\begin{aligned} w'\,(\omega) &= (1 - \frac{max_{A(\omega)} - min_{A(\omega)}}{max_{A(\omega)}})\, min_{A(\omega)} \\ &= \frac{min_{A(\omega)}^2}{max_{A(\omega)}}, \end{aligned}$$

$$\begin{aligned} max_{A(\omega)} &= \max\left({}^{G}A\,(\omega)\ ,\ {}^{C}A\,(\omega)\right)\,, \\ min_{A(\omega)} &= \min\left({}^{G}A\,(\omega)\ ,\ {}^{C}A\,(\omega)\right) \end{aligned} \tag{11}$$

and $\mathcal{F}_v$ being the set of all valid frequencies. Of course, to prevent ambiguities it is only possible to compute delays up to $\frac{\pi}{\omega}$, which is half the period of the respective frequency. Converting the difference of the sensor specific phases to time we yield following weighted time difference

$$\delta t_{G2C} = \sum_{\omega \in F_v} w\,(\omega) \, \frac{\left(k_\omega 2\pi \,+\, {}^{G}\varphi\,(\omega) - {}^{C}\varphi\,(\omega)\right)}{\omega} \tag{12}$$

where k_ω is the factor which brings the respective frequency in the range of the delay. These factors have to be computed in a second iteration. A manually chosen maximum delay is used to pick all the valid frequencies with a period smaller than this threshold. Based on the phases and amplitudes of these frequencies the time delay is computed and the factors k_ω for the rest of the frequencies are estimated. Now all frequencies can be used to refine the estimate. In our experiments we compare also an alternative where we rely only on the most significant phase shift corresponding to the frequency bin with the maximum weight $\max\left(w\left(\omega\right)\right)$. This method is based on the assumption that less noise is involved in the estimate, even though it should be less robust.

A conventional Fast or Discrete Fourier Transformation (FFT, DFT) does not allow for gaps in the signal. A generalization of the DFT which can also deal with such gaps is presented in [23] and is called Extended Discrete Fourier Transform (EDFT):

$$\mathcal{F}_\alpha\left(\omega\right) = \sum_{k=0}^{K-1} x\left(kT\right)\, \alpha\left(\omega{,}kT\right) \tag{13}$$

where, in general, $\alpha\left(\omega{,}kT\right) \neq e^{-j\omega kT}$. It is an iterative algorithm which tries to find a transform basis function which is applicable to a band-limited signal registered in a finite time interval and providing the results as close as possible to the Fourier transform. Based on this transformation we can compute the magnitudes and the phases even for sequences with gaps.

III. Spatial Alignment

The spatial registration of a camera and a gyroscope can be seen as solving the rotational part of the well-known hand-eye calibration problem. To compute the rotational alignment ${}^G\boldsymbol{R}_C$ between the gyros and the camera, we compute the relative rotations, ${}^G\boldsymbol{R}_{\Delta_i}$ and ${}^C\boldsymbol{R}_{\Delta_i}$, between two consecutive images $i-1$ and i. To get the rotations of the gyroscopes, measured in their coordinate frame $\mathcal{G}$, we simply integrate the measured rotational speed between two camera time-stamps. The rotations between two consecutive images can be computed by conventional techniques, depending on the landmarks and the knowledge of the environment, and are relative to the camera frame $\mathcal{C}$.

The task is now to find the rotation ${}^G\boldsymbol{R}_C$ which rotates the measurements from frame $\mathcal{C}$ to frame $\mathcal{G}$ according to the well-known hand-eye calibration equation

$${}^G\boldsymbol{R}_{\Delta_i} = {}^G\boldsymbol{R}_C\, {}^C\boldsymbol{R}_{\Delta_i}\, {}^G\boldsymbol{R}_C^T\,. \tag{14}$$

To solve for the best fitting spatial alignment ${}^G\tilde{\boldsymbol{R}}_C$ following least-squares problem over all camera measurements N has to be solved:

$${}^G\tilde{\boldsymbol{R}}_C = \arg\max_{\boldsymbol{R}} \sum_{i=1}^{N} \mathrm{tr}\left({}^G\boldsymbol{R}_{\Delta_i}\, \boldsymbol{R}\, {}^C\boldsymbol{R}_{\Delta_i}^T\, \boldsymbol{R}^T\right) \quad | R \in SO(3) \tag{15}$$

A closed form solution for the rotation estimation is described and derived in [22] and can be computed as follows. Let ${}^C\hat{\boldsymbol{p}}_{\Delta_i}$, ${}^G\hat{\boldsymbol{p}}_{\Delta_i}$ and ${}^G\hat{\boldsymbol{p}}_C$ denote the real eigenvectors to the eigenvalue 1 of ${}^C\boldsymbol{R}_{\Delta_i}$, ${}^G\boldsymbol{R}_{\Delta_i}$ and ${}^G\boldsymbol{R}_C$ respectively. Hence, they represent the rotation axes of these DCMs. Using this representation, the system of linear equations to solve for ${}^G\boldsymbol{p}_C$, consisting of ${}^G\hat{\boldsymbol{p}}_C$ and ${}^G\theta_C$ according to Eq. 3, can be set up by all measurement pairs as follows

$$\left[{}^C\hat{\boldsymbol{p}}_{\Delta_i} + {}^G\hat{\boldsymbol{p}}_{\Delta_i}\right]_\times {}^G\boldsymbol{p}'_C = {}^C\hat{\boldsymbol{p}}_{\Delta_i} - {}^G\hat{\boldsymbol{p}}_{\Delta_i}\,, \tag{16}$$

with $[\cdot]_\times$ denoting the skew symmetric matrix of a vector.

The Angle-Axis form of the rotation ${}^G\boldsymbol{R}_C$ can then be computed from ${}^G\boldsymbol{p'}_C$ by

$${}^G\hat{\boldsymbol{p}}_C = \frac{{}^G\boldsymbol{p}'_C}{\|{}^G\boldsymbol{p'}_C\|} \quad \text{and} \quad {}^G\theta_C = 2\tan^{-1}(\|{}^G\boldsymbol{p'}_C\|)\,. \tag{17}$$

If all rotations are measured about the same axis or the sensor coordinate frames are rotated by 180°, this system is singular and there is no unique solution. However, if there are at least two rotations about different axes and the angle between the sensors is not 180°, a unique solution exists. We detect the 180°-exception by inspecting the singular values σ_1, σ_2 and σ_3 of $\left[{}^C\hat{\boldsymbol{p}}_{\Delta_i} +^G \hat{\boldsymbol{p}}_{\Delta_i}\right]_\times$, where $\sigma_1 \geqslant \sigma_2 \geqslant \sigma_3$. If $\frac{\sigma_3}{\sigma_2} < k$, where k denotes a threshold which defines the stack of skew symmetric matrices to be rank-deficient, a rotation close to 180° between the sensor frames is expected. In this case the angular alignment is estimated using Sequential Quadratic Programming (SQP) optimization [24], where the objective function is defined as

$$\mathrm{c}_{SQP}\left({}^G\boldsymbol{R}_C\right) = \sum_{i=1}^{N_C} \|{}^G\boldsymbol{R}_C\, {}^C\hat{\boldsymbol{p}}_{\Delta_i} - {}^G\hat{\boldsymbol{p}}_{\Delta_i}\|\,. \tag{18}$$

As starting point for the estimation we use

$${}^G\boldsymbol{R}_C = \mathrm{DCM}\left(\tilde{\boldsymbol{p}}_{\Delta_i}, 180°\right)\,, \tag{19}$$

where

$$\tilde{\boldsymbol{p}}_{\Delta_i} = \frac{\tilde{\boldsymbol{p}}'_{\Delta_i}}{\|\tilde{\boldsymbol{p}}'_{\Delta_i}\|} \quad \text{and} \quad \tilde{\boldsymbol{p}}'_{\Delta_i} = \mathrm{med}\left(\left\{{}^C\hat{\boldsymbol{p}}_{\Delta_i} + {}^G\hat{\boldsymbol{p}}_{\Delta_i}\right\}_{i=1}^{N_C}\right)\,. \tag{20}$$

Hence, $\tilde{\boldsymbol{p}}_{\Delta_i}$ denotes the normalized element-wise median of the sum of rotation axes, which should be quite close to the global optimum and, thus, prevent SQP from finding local ones.

In the context of gyro-camera calibration, the closed-form solution needs some adaptions to achieve adequate robustness. In [22], critical factors affecting the accuracy and robustness have been discussed and it has been observed, that the errors are proportional to the magnitude of the measured rotations. In the case of camera to gyroscope calibration, the sensitivity for errors due to relative small inter frame rotations becomes crucial and may lead to wrong results. Further, Eq. 16 minimizes the steady measurement error of a conventional robot-camera setup, but it does not consider any time correlation between the sensors. It is necessary to overcome this problem if we want to apply this method for bias prone gyroscopes.

Small rotations suffer from a small signal to noise ratio and a large discrepancy between the sensor measurements implies an erroneous sample pair. If Eq. 16 is applied without any modification, each inter frame rotation would affect the outcome equally - independent of whether there is no rotation, and the measurement consists only of noise, or in presence of an outlier. Therefore, the absolute angles of the inter frame rotations should be used to weight the respective equation. This leads to following weights for both sides of Eq. 16 and the summands in Eq. 18

$$w_i = (1 - \frac{max_{\theta_i} - min_{\theta_i}}{max_{\theta_i}})\, min_{\theta_i} = \frac{min^2_{\theta_i}}{max_{\theta_i}} \qquad (21)$$

with max_{θ_i} and min_{θ_i} analogously defined to Eq. 11.

Calibration sequences are usually short and, therefore, a common assumption is that the IMU biases are constant. We estimate the bias using a Levenberg-Marquardt optimization. In the experiments we compare following two cost functions. In the first approach we simply try to find the bias by making the absolute angles as similar as possible, while disregarding outliers. This is achieved by applying the Blake-Zisserman cost function

$$\mathrm{c}_{BZ}(\delta) = -\ln\left(e^{-\delta^2} + \epsilon\right) \qquad (22)$$

with the crossover point from inliers to outliers given by the threshold α in $\epsilon = e^{-\alpha^2}$ [25]. Thus, this optimization problem may be written as

$$\tilde{\boldsymbol{b}} = \arg\min_{\boldsymbol{b}} \sum_{i=1}^{N} \mathrm{c}_{BZ}\left({}^G\theta'_{\Delta_i} - {}^C\theta_{\Delta_i}\right) \qquad (23)$$

with ${}^G\theta'_{\Delta_i}$ being the absolute angle corresponding to

$${}^G\boldsymbol{R}'_{\Delta_i} = \prod_{t=t_{i-1}}^{t_i} {}^G\boldsymbol{R}'_t \qquad (24)$$

where

$${}^G\boldsymbol{R}'_t = \mathrm{DCM}\left({}^G\overline{m}_T\left(\begin{bmatrix} {}^G\dot{\phi}_t \\ {}^G\dot{\chi}_t \\ {}^G\dot{\psi}_t \end{bmatrix} - \boldsymbol{b}\right)\right) \qquad (25)$$

and $\tilde{\boldsymbol{b}}$ being the estimated bias.

An other approach is to minimize the trace of the covariance matrix, $\boldsymbol{P}_{{}^G\boldsymbol{p}_C}$, of ${}^G\boldsymbol{p}_C$. The covariance matrix for ${}^G\boldsymbol{p}'_C$ results from the over-constrained linear system (see Eq. 16) as

$$\boldsymbol{P}_{{}^G\boldsymbol{p}'_C} = \boldsymbol{V}_{[\cdot]_\times}\boldsymbol{\Sigma}^{-2}_{[\cdot]_\times}\boldsymbol{V}^T_{[\cdot]_\times} \qquad (26)$$

with

$$\boldsymbol{U}_{[\cdot]_\times}\boldsymbol{\Sigma}_{[\cdot]_\times}\boldsymbol{V}^T_{[\cdot]_\times} = \mathrm{SVD}\left(\boldsymbol{B}^{C+G}_{[\cdot]_\times}\right) \qquad (27)$$

and $\boldsymbol{B}^{C+G}_{[\cdot]_\times}$ representing the stack of the skew matrices of all the vector sums, $\left[{}^C\hat{\boldsymbol{p}}_{\Delta_i} + {}^G\hat{\boldsymbol{p}}_{\Delta_i}\right]_\times$. Depending on the length of the calibration sequence, this matrix may become quite large. To reduce the processing time one can also use

$$\boldsymbol{P}_{{}^G\boldsymbol{p}'_C} = \boldsymbol{V}_T\left(\mathrm{tr}\left(\boldsymbol{\Sigma}^2_T\right)\boldsymbol{I} - \boldsymbol{\Sigma}^2_T\right)^{-1}\boldsymbol{V}^T_T \qquad (28)$$

with $\boldsymbol{U}_T\boldsymbol{\Sigma}_T\boldsymbol{V}^T_T = \mathrm{SVD}\left(\boldsymbol{B}^{C+G}_T\right)$ and $\boldsymbol{B}^{C+G}_T$ being the row-wise stack of the vectors $\left({}^C\hat{\boldsymbol{p}}_{\Delta_i} + {}^G\hat{\boldsymbol{p}}_{\Delta_i}\right)^T$. A derivation of this equation is not shown due to lack of space.

Minimizing the covariance matrix of ${}^G\boldsymbol{p}'_C$ does not necessarily mean to minimize the covariance of ${}^G\boldsymbol{p}_C$. Therefore, we propagate the covariance and the estimate for ${}^G\boldsymbol{p}'_C$ through the nonlinear equation

$${}^G\boldsymbol{p}_C = \frac{2\,{}^G\boldsymbol{p}'_C}{\sqrt{1 + |{}^G\boldsymbol{p}'_C|^2}} \qquad (29)$$

by using Sigma-Points as, *e.g.*, in the Unscented Kalman filter [26]. Sticking to the notation of that paper, the weight $W^{(0)}$ is set to 0 according to the formula which computes the optimum assuming a Gaussian distribution

$$W^{(0)} = 1 - \frac{N_S}{3} \qquad (30)$$

with N_S denoting the vector length which is three in our case. Thus, the objective function is this time

$${}^G\tilde{\boldsymbol{R}}_C = \arg\min_{\boldsymbol{b}} \left(\mathrm{tr}\left(\boldsymbol{P}_{{}^G\boldsymbol{p}_C}\right)\right). \qquad (31)$$

In the next Section we will compare the presented methods based on simulated and real data.

IV. Experiments

In our experiments we used simulated data and real data acquired by an IMU-camera pair as illustrated in Fig. 1. The camera acquires images with 15 Hz while the IMU has a sampling frequency of 120 Hz. To verify our methods we acquired two sets of four runs each. The first set of runs is approximately 10 seconds long each and the latter four runs last about 40 seconds each. The camera rotation has been computed by extracting the corners of a checkerboard and estimating the camera position in an optimization framework using Calde and Callab [27]. We neglect the error of the image-based rotation estimation which is due to the correlation between rotation and translation. This correlation will be considered in a subsequent full spatial registration. Otherwise, one can also use translation invariant (far distant) landmarks to estimate the rotation uncoupled from the translation [28]. First we will address the temporal alignment and after that we show some experimental results for the rotation estimation.

A. Temporal Alignment

Before the measurements can be aligned, the time-stamps have to be fixed as described in Section II. Therefore, the sampling period has to be estimated and sample jams and gaps have to be detected as illustrated in Fig. 2.

In the following, we want to show the advantages and drawbacks of the temporal methods discussed in Section II. We tested the delay estimation methods with simulated data based on a weighted overlap of nine sine frequencies on all axes, whereas the frequencies are in between 0.1 Hz and 90 Hz. The sampling period of the simulated gyroscopes is 10 ms and of the camera 40 ms. As window function for the EDFT in the phase congruency method we chose a Hamming window, which is a good trade-off between high dynamic-range and sensitivity. For the maximum weight phase shift estimation we used the rectangular window, because it provides the highest sensitivity.

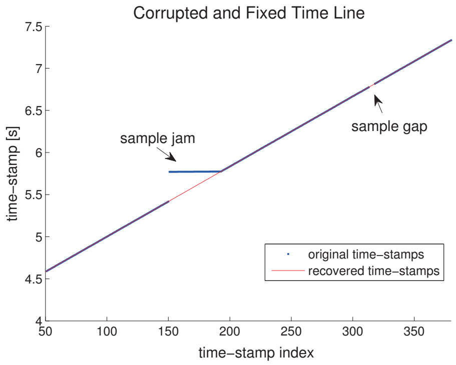

Fig. 2. The blue dots represent the time-stamps of a clipped image sequence. The red line shows the fixed-time line without data jams and gaps.

Fig. 3 illustrates a simulated delay between IMU and camera of 0.5 s and the corresponding cross correlation result. The delay is rather long compared to real world cases, but it has been chosen to show the limits of the correlation method. Fig. 4 depicts the estimated magnitudes and the estimated phases for both sensors. The phases of corresponding peaks are used for temporal alignment.

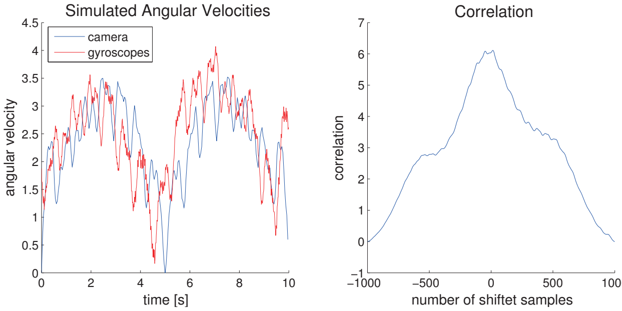

Fig. 3. The left figure illustrates the unaligned simulated gyroscope (red) and camera data (blue), while the right figure shows the cross correlation of the data.

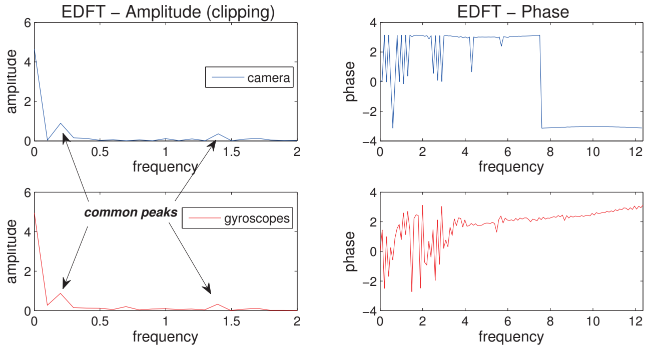

Fig. 4. This figures illustrate the phase shift based temporal alignment for the simulated data. The left images show the amplitude and the right the phase spectrum of the gyroscope (red) and camera (blue) data.

The result of the average phase shift based approach is 0.5193 s, while the phase shift of the most significant frequency yields 0.5344 s. The correlation-based approach estimates a 0.14 s time lag. The explanation for this can be found in Fig. 5. There we evaluated the performance on various delays. While the cross-correlation works highly accurate and reliable for small delays, it finds a wrong optimum for large delays. The average and maximum phase shift methods seem to be less accurate but more robust as the delay increases. The difference between these two methods becomes apparent when comparing the result with the real data, as done in Fig. 6. Following table summarizes some statistical quantities of the outcome:

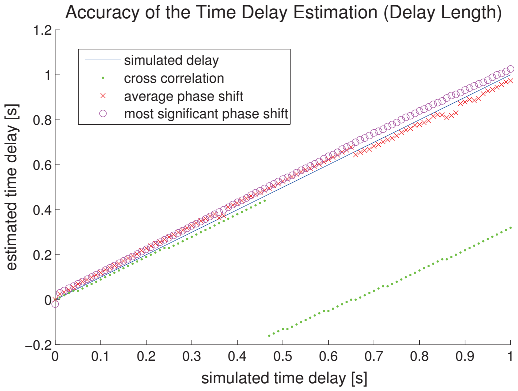

Fig. 5. The blue line represents the simulated delay. The green dots are the correlation based results. The red crosses and the magenta circles are the average and the most significant phase shift output respectively.

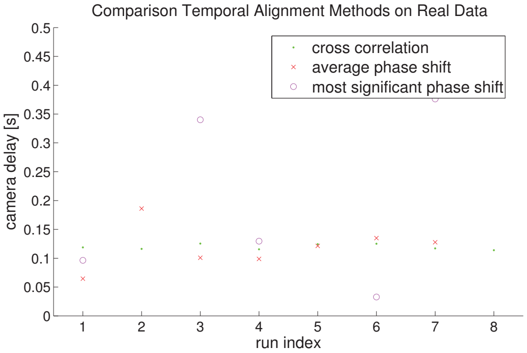

Fig. 6. The cross-correlation reliably estimates the time delay, while the phase congruency methods are not as robust. Some of their estimates are not even visible because of the scaling which has been chosen for sake of detail.

method - unit [s]	*mean*	*median*	*std. dev.*	*range*
cross corr.	0.1195	0.1179	0.0047	0.0115
av. phase shift	0.1831	0.1246	0.1841	0.5659
most sig. ph. sh.	0.4870	0.2348	0.8704	2.7979

While the cross correlation method achieves coherent results in all eight runs, the phase based approaches proof to be not as reliable for real application. This is because white noise affects all frequencies and, hence, the phase shift can not be used for accurate temporal alignment of noisy data.

B. Rotational Alignment

To evaluate the noise sensitivity of the presented rotation estimation we added white Gaussian noise to the simulated data. Fig. 7 shows the performance of the weighted and the unweighted rotation estimation. The weighted rotation estimation always outperforms the unweighted method.

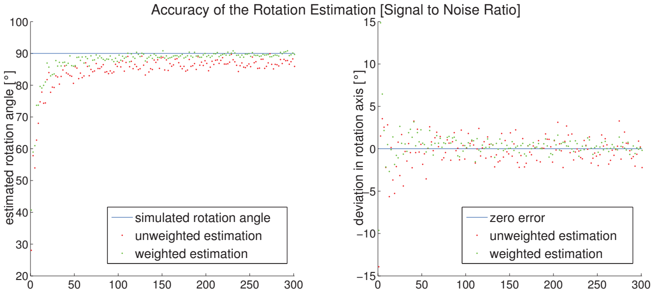

Fig. 7. The left image shows the results for the absolute rotation angle, the right picture illustrates the angular error between the estimated and simulated rotation axis. The blue line represents the simulated absolute angle and the zero-error line respectively. By increasing the signal to noise ratio the performance of both, the weighted (green) and the unweighted (red) rotation estimation, increases.

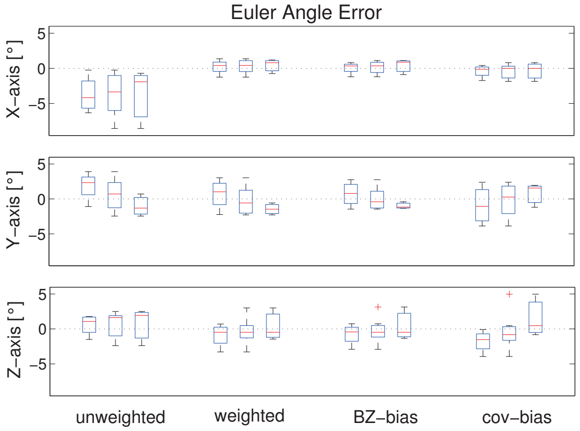

Fig. 8. These box plots show the performance of the presented rotational alignment methods. Therefore, the estimated Euler angles between IMU and camera have been substracted from a "ground truth" estimate gathered by a brute force search with resolution $0.5°$. The leftmost box plot of each set corresponds to the four short runs, the rightmost to three of the long runs and the center box plot represents all seven runs.

It is difficult to determine an accurate ground truth for the rotational alignment between a camera and an IMU without having a CAD drawing of the setup. Therefore, we performed a brute force search with a resolution of $0.5°$ to find a sort of "ground truth". As objective function we used Eq. 15. Unfortunately, in one of the runs a local minimum far off any possible solution has been found which proofs the non-convexity of the problem and, hence, that filtering and optimization techniques may easily find a local instead of the global minimum if the starting point is far off the actual solution. This run has not been considered in Fig. 8, which shows the deviation of the various methods described in Section III relative to the brute-force estimate. The estimates of the unweighted (original) and the weighted rotation are optimized according to Eq. 16 and not after the nonlinear Eq. 29 as it should be. Thus, we propagate the results by the unscented transformation described in Section III, with the outcome, that the difference between the propagated and original values is negligible and, thus, the unscented transformation can be spared. Such a transformation becomes only necessary if the covariance matrix is large, which is not the case for our data. The weighted approach proves to be more reliable and accurate compared to the original hand-eye calibration technique. Further, the bias estimates are more coherent with the weighting method than the covariance minimization approach.

In our last experiments we want to underline the importance of a proper temporal alignment. It is difficult to argue based on simulations which temporal accuracy is relevant for spatial alignment, because it depends strongly on the dynamics of the registration run, the noise of the sensors, the accuracy of the camera based pose estimation, and so on. Therefore, we run the rotation estimation on misaligned real data. The time delays have been chosen between the mean estimate of all runs and a delay of zero, which means that no temporal alignment is provided. Fig. 9 compares the different runs. While the median does only change a little for the different time delays, the variance increases significantly for bad aligned data. Further, the plot shows that the weighting of the data samples also increases the robustness against temporal misalignment.

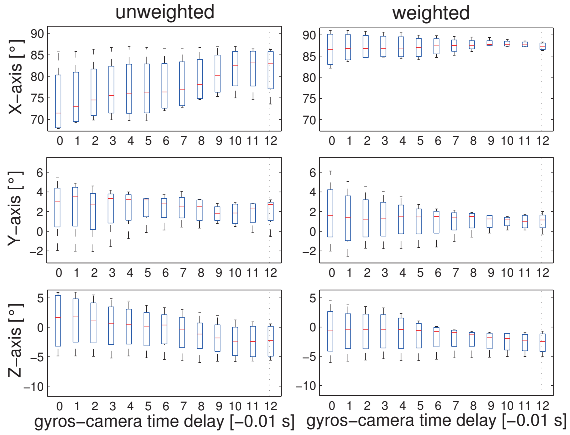

Fig. 9. These box plots show the performance of the unweighted and weighted rotation estimation on not correctly temporally aligned real data. The black dotted line corresponds to the mean of the estimated delays between camera and gyros of all runs.

The batch-based nonlinear optimization described in [13] also shows to be sensitive to proper temporal alignment. This approach models the sensors' trajectory as B-spline and optimizes for the spatial alignment. To allow for natural landmarks also the scale of translation α is estimated. In our experiments we know the proper scale of the checkerboard and, thus, also of the translation. Hence, in our experiments $alpha$ should always be one. However, this factor has shown to be rather sensitive to the spatial or temporal alignment and, therefore, it is used as index for the quality of the optimization result. In this experiment we varied the estimated time delay from $0\,\%$ to $200\,\%$ and plotted the box plots of the estimates for the translational scale to evaluate the effect of a time lag between the measurements. The result worsens significantly if the measurements are not aligned correctly (100 %). This experiment stresses the importance of proper temporal alignment also for complex registration methods. Adding the delay between the IMU and camera measurement as parameter to $\boldsymbol{\theta}$ would probably worsen the convergence properties of the optimization and make the analytical calculation of the Jacobian significantly more complex.

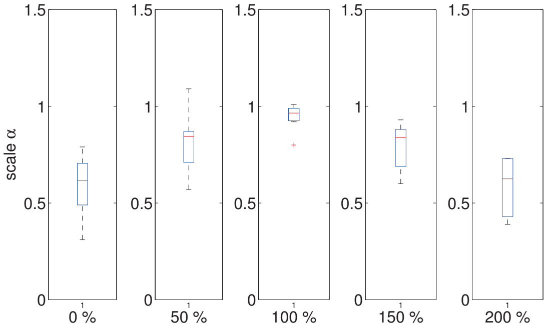

Fig. 10. Estimated scale factor α with respect to the temporal alignment: 0 %, no temporal alignment; 100 %, correct temporal alignment; 200 %, twice the estimated delay between camera and IMU.

V. Conclusion

In this work we first motivated the importance of proper temporal alignment of inertial-visual systems. Temporal alignment is crucial for reliable fusion of any sensor data, especially if involving sensors which demand high dynamics for a feasible signal to noise ratio. We compared two approaches, with the outcome that cross-correlation is the better choice for real data with the premise of a reasonable short delay. Our time delay estimation does not require a spatial alignment of the sensors and, thus, prevents a possible compensation of the error in the spatial estimate by the computed time delay. Further, by assuming a constant sample period, we are able to deal with random jitter, missing samples and data jams in the data pre-processing.

We derived a method from hand-eye calibration, robustified it for small and error-prone rotations and proposed an exception handling for ill-posed configurations. The robustness and reliability of different alternatives has been compared in experiments on simulated and real data. According to these, the best choice is to use the weighted angular alignment and the bias estimation based on Eq. 23. The presented approach is easy to apply and does not contain many parameters and degrees of freedom, which makes the filter or optimization based methods complex and error-prone. Even though it is not guaranteed that the closed form solution finds the global optimum due to the biases in the IMU measurements, it is quite probable to converge to it as soon as the biases are estimated within the optimization framework.

However, in case that also the translational offset should be estimated, the acquired results may be used as starting point for any filter or optimization based approach.

VI. Acknowledgments

This work was partially funded by the DLR internal project for image-based navigation systems.

References

[1] K.H. Strobl, E. Mair, T. Bodenmüller, S. Kielhöfer, W. Sepp, M. Suppa, D. Burschka, and G. Hirzinger. The self-referenced dlr 3d-modeler. In *IEEE/RSJ IROS*, October 2009.

[2] C.N. Taylor. Fusion of inertial, vision, and air pressure sensors for mav navigation. In *IEEE Multisensor Fusion and Integration for Intelligent Systems*, Aug 2008.

[3] A. Chilian and H. Hirschmuller. Stereo camera based navigation of mobile robots on rough terrain. In *IEEE/RSJ IROS*, oct 2009.

[4] E. Mair, K.H. Strobl, T. Bodenmüller, M. Suppa, and D. Burschka. Real-time image-based localization for hand-held 3d-modeling. *Künstliche Intelligenz*, 24, May 2010.

[5] J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. In *IEEE ICRA Workshop on Integration of Vision and Inertial Sensors - 2nd InerVis*, Apr 2005.

[6] J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *International Journal of Robotic Research*, 26(6):561575, jun. 2007.

[7] F.M. Mirzaei and S.I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration. In *IEEE/RSJ IROS*, pages 2427 –2434, oct. 2007.

[8] J. Kelly and G.S. Sukhatme. Fast relative pose calibration for visual and inertial sensors. In *Proc. 11th Int. Symposium Experimental Robotics*, July 2008.

[9] J. Kelly and G.S. Sukhatme. Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration. In *Proceeding IEEE Int. Symp. Computational Intelligence in Robotics and Automation*, pages 360–368, Dec 2009.

[10] F.M. Mirzaei and S.I. Roumeliotis. *IMU-Camera Calibration: Bundle Adjustment Implementation*, aug. 2007.

[11] J. D. Hol, T. B. Schön, and F. Gustafsson. Modeling and calibration of inertial and vision sensors. *Int. Journal of Robotic Research*, 29, February 2010.

[12] P. Lang and A. Pinz. Calibration of hybrid vision / inertial tracking systems. In *2nd Workshop on Integration of Vision and Inertial Sensors*, Apr 2005.

[13] M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka. Optimization based imu camera calibration. In *IEEE/RSJ IROS*, 2011.

[14] B. Simons. An overview of clock synchronization. *Fault-Tolerant Distributed Computing*, pages 84–96, 1990.

[15] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *Network, IEEE*, 18(4):45–50, 2004.

[16] T. Lei and CA Stelios. Decentralized filtering with random sampling and delay. *Information Sciences*, 81(1-2):117–131, 1994.

[17] T.D. Larsen, N.A. Andersen, O. Ravn, and N.K. Poulsen. Incorporation of time delayed measurements in a discrete-time kalman filter. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 4, pages 3972–3977. IEEE, 1998.

[18] W. Li and H. Leung. Simultaneous registration and fusion of multiple dissimilar sensors for cooperative driving. *Intelligent Transportation Systems*, 2004.

[19] S.J. Julier and J.K. Uhlmann. Fusion of time delayed measurements with uncertain time delays. In *American Control Conference, 2005. Proceedings of the 2005*, pages 4028–4033. IEEE, 2005.

[20] F. Tungadi and L. Kleeman. Time synchronisation and calibration of odometry and range sensors for high-speed mobile robot mapping. In *ACRA08*, 2008.

[21] J. Kelly and G.S. Sukhatme. A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors. In *IFRR ISER'10*, 2010.

[22] R.Y. Tsai and R.K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. In *IEEE Transactions on Robotics and Automation*, 1989.

[23] V.Y. Liepin'sh. An algorithm for evaluation a discrete fourier transform for incomplete data. *Automatic control and computer sciences*, 30(3):27–40, jun. 1996.

[24] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 2000.

[25] A. Blake and A. Zisserman. *Visual reconstruction*. MIT Press, 1987.

[26] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 2004.

[27] K.H. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter. DLR CalDe and DLR CalLab.

[28] Elmar Mair and Darius Burschka. *Mobile Robots Navigation*, chapter Zinf - Monocular Localization Algorithm with Uncertainty Analysis for Outdoor Applications, pages 107 – 130. In-Tech, 2010.