Video-based realtime IMU-Camera Calibration for Robot Navigation

Arne Petersen and Reinhard Koch

Christian-Albrechts-Universität, D-24098 Kiel, Germany

ABSTRACT

This paper introduces a new method for fast calibration of inertial measurement units (IMU) with cameras being rigidly coupled. That is, the relative rotation and translation between the IMU and the camera is estimated, allowing for the transfer of IMU data to the cameras coordinate frame. Moreover, the IMUs nuisance parameters (biases and scales) and the horizontal alignment of the initial camera frame are determined. Since an iterated Kalman Filter is used for estimation, information on the estimations precision is also available. Such calibrations are crucial for IMU-aided visual robot navigation, i.e. SLAM, since wrong calibrations cause biases and drifts in the estimated position and orientation. As the estimation is performed in realtime, the calibration can be done using a freehand movement and the estimated parameters can be validated just in time. This provides the opportunity of optimizing the used trajectory online, increasing the quality and minimizing the time effort for calibration. Except for a marker pattern, used for visual tracking, no additional hardware is required.

As will be shown, the system is capable of estimating the calibration within a short period of time. Depending on the requested precision trajectories of 30 seconds to a few minutes are sufficient. This allows for calibrating the system at startup. By this, deviations in the calibration due to transport and storage can be compensated. The estimation quality and consistency are evaluated in dependency of the traveled trajectories and the amount of IMU-camera displacement and rotation misalignment. It is analyzed, how different types of visual markers, i.e. 2- and 3-dimensional patterns, effect the estimation. Moreover, the method is applied to mono and stereo vision systems, providing information on the applicability to robot systems. The algorithm is implemented using a modular software framework, such that it can be adopted to altered conditions easily.

Keywords: Kalman Filter, inertial measurement units, IMU-camera calibration, IMU-aided visual SLAM

1. INTRODUCTION

Nowadays autonomous, mobile robot systems become increasingly important. This is due to the fact, that the cost of manufacture decreased dramatically during the past decades. Moreover, by now they are applicable to versatile environments (ground and air,[1] underwater[2]) and can be equipped with various types of sensors (ultrasonic,[3] laser scanners[4] etc.). These devices sense the environment and, by this, provide the information needed for self planed and self supervised navigation. To allow for route planing and other basic navigation tasks, two problems have to be solved in advance. On the one hand the systems has to be able to localize itself. On the other hand it has to build up a map of the environment. Since the localization is done in the environment map and the map is build up using the estimated poses (“localizations”) these tasks has to be solved simultaneously. This procedure is called **S**imultaneous **L**ocalization **A**nd **M**apping or SLAM for short.

Applying SLAM to multi-sensor systems requires fusion of all acquired sensor data. When information from independent sensors is to be merged, it has to be available within a common coordinate system. This fact is well known from applications in computer vision, where more than one camera is used. Stereo camera rigs have to be calibrated, to allow for the reconstruction of depth maps or 3D-models of the observed scene. Our work aims at the fusion of visual and inertial data, i.e. localization and mapping shall be done using imaging systems aided by **I**nertial **M**easurement **U**nits, IMUs for short. When speaking of aiding, it is meant, that the estimation of the localization and mapping is done using the images solely. The IMU aids this process, by providing precise predictions for the system movement between two captured images. This way, initial values of high quality can be provided to the pose and map estimation algorithms. Estimators using such predictions are known to not only be more robust to fast movements but also to be much more precise.

The challenge, when using IMU data to predict camera movements, is to compensate the misalignments between the sensors. I.e. when they are related by unknown translations and rotations, the IMU data cannot be used to predict the camera movement. Due to this fact it is essential to determine the calibration, needed to transfer the IMU data to the cameras coordinate system.

Previous Work In the past years some research has been made, to solve the task of IMU-camera calibration. Lobo[5] et al. describe an algorithm to estimate the IMU-camera calibration in two steps. At first only the rotational alignment is estimated and following that only the displacement. Thus, correlations between these transformation estimations are ignored. Moreover they make use of additional hardware ("horizontal" features and turntables). Other systems (Mirzaei[6] et al., Hol[7] et al, Kelly[8] et al.) only require a calibration pattern, that has to be horizontally aligned. The algorithm presented by Mirzaei[6] seems to produce good estimation results. But it is not realtime capable and the initialization of the alignment to be estimated was given rather precise (5cm and $< 1°$). While Hol[7] and Kelly[8] introduced realtime capable calibration systems, they performed tests also only for rather small misalignments. The deviations between the initialization misalignment and their estimate are at most 4cm and $1°$ and 5cm and $8°$ respectively. Since Hol[7] propose an algorithm designed for fish-eye cameras, it cannot be applied to standard cameras, i.e. to stereo rigs.

Contribution In this paper we propose an algorithm using an iterated extended Kalman Filter[9] providing an IMU-camera calibration in realtime using a simple marker only. In contrast to the existing methods this marker has not to be horizontally aligned. Moreover it is able to cope with large initialization errors and rather short trajectories. An implementation is provided allowing for online verification of the estimation process using visualization techniques. This way the trajectories traveled can be adopted to the estimation quality reached so far.

In this paper at first the problem of IMU-camera calibration is defined precisely. Afterwards a solution is proposed and some experiments are discussed. Finally we conclude our work.

2. PROBLEM STATEMENT

2.1 Notation

In the following different coordinate systems and transformations between these frames will be used. To ease the readability of such equations a simplified notation is given. Entities represented in Euclidean 3-space are given with respect to a coordinate frame. Let a a coordinate frame and t a translation in Euclidean space. The notation t^a is meant to represent the translation in coordinates of a. Moreover the lower indices in t^a_{ab} indicate, that the translation from a to b is referenced.

The basic transformation between two coordinate frames a and b is a combination of a rotation and a translation. Let R describe the rotation between a and b having the same coordinate origin. By R^b_a the rotation is given transferring a point p given in frame a to frame b. That is:

$$p^b \quad = \quad R^b_a \cdot p^a \quad \text{and} \quad p^a \quad = \quad {R^b_a}^T \cdot p^b \quad = \quad R^a_b \cdot p^b \tag{1}$$

Since the models defined later use Euler angles ϕ^b_a to represent a rotation matrix R^b_a, these are used interchangeable in the following. When a and b differ also in the coordinate origin, the translation t_{ab} from a to b has to be taken into account. That is:

$$p^b \quad = \quad R^b_a \cdot p^a - t^b_{ab} \quad = \quad R^b_a \cdot (p^a - t^a_{ab}) \quad = \quad R^b_a \cdot (p^a + t^a_{ba}) \tag{2}$$

When transferring entities between coordinate frames, relative entities like translations, velocities, accelerations etc. have to be treated specially. Since they are independent of a translation in space, they are transformed using the rotation only. Moreover, the lower indices are interpreted differently. For example the velocity v_{ab} describes the velocity of b with respect to a.

2.2 IMU-Camera Calibration

In IMU aided visual SLAM the inertial sensors are mostly used to drive an inertial motion model by providing the velocity increments (accelerations) and the angle increments (turn rates). The motion model integrates the accelerations and turn rates to determine the changes in the systems pose, where the pose is defined to describe the systems position and orientation.

The poses are given relative to a navigation coordinate frame. For our description we need additional frames, to describe the used entities. These systems are:

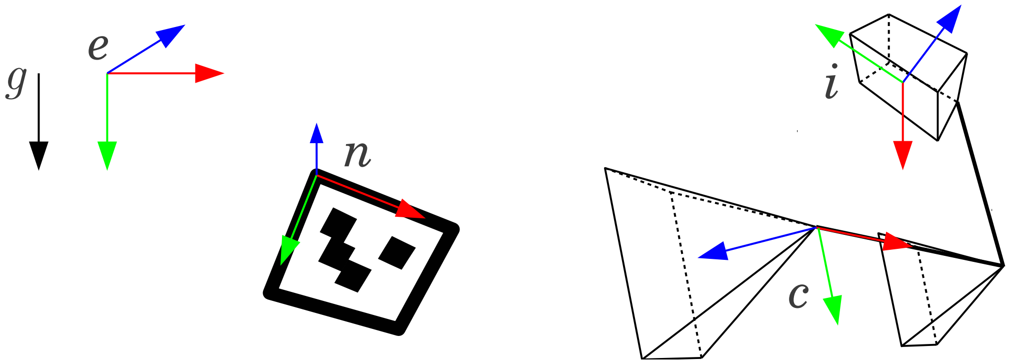

Figure 1. Visualization of coordinate frames and gravity vector g.

The world coordinate system e being horizontally aligned to the earths surface. That is, its Y-axis heads towards the earths center and thus is parallel to the earths gravity. Since an alignment to north direction cannot be determined, the heading (rotation around Y-axis) is given by the cameras initial orientation.

The navigation frame n is determined by the position and orientation of the marker pattern, used for visual tracking. It will be used to describe the cameras pose. Moreover, the movement model, providing the systems motion prediction, is applied in this frame.

The camera frame c is the cameras body fixed coordinate system. The calibration parameters to be estimated are given relative to this frame.

The IMUs frame i is the IMUs body fixed coordinate system. The data provided by the inertial sensor relates to this frame.

For clearness these are visualized in figure 1. For visual SLAM the relation between the camera frame and the navigation frame is to be estimated. This relation is called system pose in the following. When using IMUs to aid the pose estimation, the frames i and e are needed in addition. This is due to the fact, that the measured accelerations also include the earths gravity, i.e. the inverse gravity acceleration. Thus, the world frame e must be known, to enable the compensation for this measurement disturbance. Because the data provided by the IMU is given in i, the relation between the IMU and the camera frame c needs to be known.

While the world frame e has to be estimated during the SLAM process, the relation of c and i can be determined in advance, to minimize the estimation complexity. Estimating the rotation R_c^i and the respective translation t_{ci} between the camera and the IMU is known as IMU-camera or hand-eye calibration. This task is to be solved by the methods proposed in this article.

2.3 Hardware Setup

The hardware used for evaluating the algorithm, consists of a stereo camera rig and a low cost IMU sensor, as visualized schematically in figure 1. Since camera and stereo calibration are out of the scope of this article, the cameras internal parameters and lense distortion along with the stereo relations are assumed to be known. For the used hardware this was achieved using the calibration framework[*] described by Schiller[10] et al. During the estimation process this information will be used to rectify the marker points detected in the image instead of rectifying the whole image.

The stereo cameras provide images with a frame rate of $15hz$ and the IMU delivers accelerations and rates of turn at $100hz$. The camera is connected to a standard desktop PC (Intel(R) Core(TM) i7 @ 2.67GHz) via an IEEE1394 interface. The IMU provides its data through a USB connection. A crucial factor for the combination of different sensors is the relation in time. That is, the timestamps for all acquired measurements have to be known. For many SLAM systems no

[*] MIP - MultiCameraCalibration, http://www.mip.informatik.uni-kiel.de/tiki-index.php?page=Calibration - visited 19th march 2012

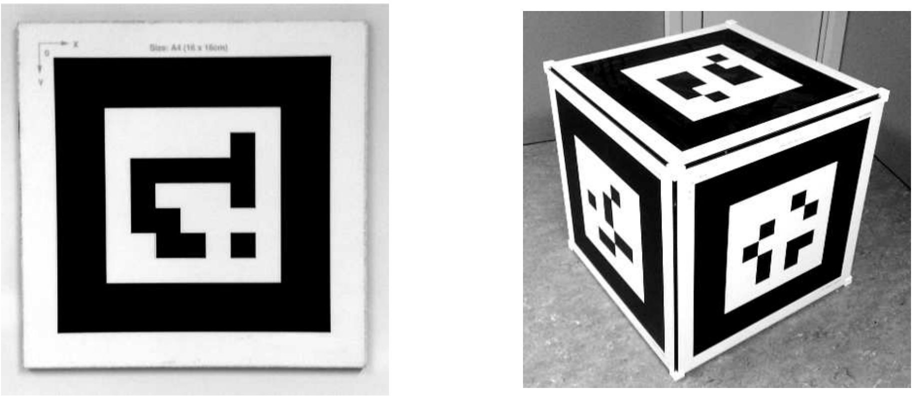

Figure 2. Marker used for pose tracking. Left: small (16x16cm) 2D-marker; right: large (60x60x60cm) 3D-marker

explicit hardware triggers are available, to ensure the measurements temporal alignment. Thus, our system was chosen to use the timestamps separately provided by the cameras and the IMU. This way our evaluation results can be applied also to systems being not triggered by hardware clocks.

For evaluating the impact of different types of calibration patterns we used two different markers. The first is a small 2D-marker (16x16 cm) and the second a large 3D-marker (60x60x60 cm). Both are visualized in figure 2.

3. MODELING AND ESTIMATION

This section provides a detailed description of the used models and how they are embedded in the Kalman-Filter. Moreover, the work flow of the calibration application is described.

3.1 System Modeling

In advance to the system models the state parametrization, to be used in the Kalman filter, has to be discussed. Because two automatically selected stages of estimation are used, two different state parametrizations are given. The system models are then divided into prediction and observation models. The former is the movement model f. It makes use of inertial measurements (α acceleration, ω rates of turn) to predict changes in the systems position and orientation. The latter is the measurement model h, relating the 3-space points represented by the markers to the image points to associated with them.

State Parametrizations: The system state p is made up of the systems pose and velocity, nuisance parameters for the IMU. Moreover, the horizontal alignment of the navigation frame n and the IMU-camera calibration are included. As already stated, we use two different state parametrizations. These differ in the coordinate frames the systems pose is given in. At first the pose is given as the relation between the camera frame c and the navigation frame n. The second stage uses the transformation between n and the IMUs frame i as system pose.

The positions t^n_{nc} and t^n_{ni}, the velocities v^n_{nc} and v^n_{ni} as well as the orientations ϕ^c_n and ϕ^i_n (given as 3 Euler-angles) are represented relative to and in the navigation coordinate frame n. The IMU nuisance parameters are given by the acceleration and turn rate biases b_α and b_ω as well as the respective scales s_α and s_ω. The horizontal alignment ϕ^n_e consisting of 2 Euler angles represents the orientation of n with respect to e. As already stated in 2.2 the heading cannot be determined. Thus, only a 2-axis rotation is estimated.

Finally the IMU-camera calibration is given by the camera to IMU translation t^c_{cb} in camera coordinates and the 3 Euler angles ϕ^i_c. By this, the state parameters for the 1. and 2. stage are:

$$p^1 = \begin{pmatrix} t^n_{nc} & v^n_{nc} & \phi^c_n & b_\alpha & b_\omega & s_\alpha & s_\omega & \phi^n_e & t^c_{ci} & \phi^i_c \end{pmatrix}^T \tag{3}$$

and

$$p^2 = \begin{pmatrix} t^n_{ni} & v^n_{ni} & \phi^i_n & b_\alpha & b_\omega & s_\alpha & s_\omega & \phi^n_e & t^c_{ci} & \phi^i_c \end{pmatrix}^T \tag{4}$$

The Prediction Models are given by standard inertial movement models. That is, first and second Taylor expansions in time domain are used to determine position, velocity and orientation changes in dependency of the time increment ΔT. All remaining parameters are assumed to be constant for the whole estimation process. For the first stage of estimation only the measured turn rates ω are used for movement prediction. In contrast, the second stage makes full use of accelerations α and turn rates ω.

For the first stage only a first order Taylor expansion for the position and orientation prediction is used. The velocity is assumed to be constant. By this, the first stage prediction models for the position f_t^1 and the orientation f_ϕ^1 result in:

$$
\begin{aligned}
f_t^1(p^1) &= t_{nc}^n + \Delta T \cdot \partial t_{nc}^n &&= t_{nc}^n + \Delta T \cdot v_{nc}^n && (5)\\
f_\phi^1(p^1) &= \phi_c^n + \Delta T \cdot \partial \phi_c^n &&= \phi_c^n + \Delta T \cdot s_\omega \cdot R_i^c \cdot (\omega + b_\omega) && (6)
\end{aligned}
$$

Where R_i^c, b_ω and s_ω are to be estimated in this first stage. For the second stage a second order expansion for the position f_t^2 and first order expansions for the velocity f_v^2 and orientation f_ω^2 are used:

$$
\begin{aligned}
f_t^2(p^2) &= t_{ni}^n + \Delta T \cdot \partial t_{ni}^n + \frac{1}{2}\Delta T^2 \cdot \partial^2 t_{ni}^n &&= t_{ni}^n + \Delta T \cdot v_{ni}^n + \frac{1}{2} \cdot \Delta T^2 \cdot \alpha_{ni}^n && (7)\\
f_v^2(p^2) &= v_{ni}^n + \partial v_{ni}^n &&= v_{ni}^n + \Delta T \cdot \alpha_{ni}^n && (8)\\
f_\phi^2(p^2) &= \phi_c^n + \partial \phi_c^n &&= \phi_c^n + s_\omega \cdot R_i^c \cdot (\omega + b_\omega) && (9)
\end{aligned}
$$

To be able to apply this movement models, the acceleration α_{ni}^n has to be derived using the IMU input α, its nuisance parameters and the earth acceleration g. This is done using:

$$
\begin{aligned}
\alpha_{ni}^n &= R_i^n \cdot \alpha_{ni}^i = R_i^n \cdot \left(s_\alpha \cdot (\alpha + b_\alpha) + g^i\right) = R_i^n \cdot \left(s_\alpha \cdot (\alpha + b_\alpha) + R_n^i \cdot R_e^n \cdot g^e\right) && (10)\\
&= s_\alpha \cdot R_i^n \cdot (\alpha + b_\alpha) + R_e^n \cdot g^e && (11)
\end{aligned}
$$

These ways of determining the true accelerations and rates of turn from the IMU data are strongly simplified. For a more precise computation effects like the earth transport rate and others have to be considered. Since the space of navigation for the calibration process is within a few meters and trajectories are not longer than a few minutes, such effects can be neglected in this case.

The Observation models are used to map the known 3-space points of the marker pattern to the image plane, exploiting the predicted pose by projective geometry (see Hartley[11] et al.). In the following X_i represent these 3-space points, $\mathcal{X}_i$ their homogeneous projection and x_i their position in the image plane. Here only $\mathcal{X}_i$ will be given, since

$$
x_i = \frac{1}{\mathcal{X}_{i(3)}} \cdot \begin{pmatrix} \mathcal{X}_{i(1)} \\ \mathcal{X}_{i(2)} \end{pmatrix} \qquad (12)
$$

holds for every projection.

As for the prediction model the observation models differ in the used coordinate frames. For the first stage (prediction in camera frame c) the estimated pose is the camera pose. Thus, the projection can be done by

$$
\mathcal{X}_i = h^1(X_i, p^1) = R_n^c \cdot (X_i - t_{nc}^n) \qquad (13)
$$

where R_n^c is the rotation matrix corresponding to the Euler angles ϕ_n^c from the system (camera) pose.

Since the system state for the second stage is parametrized in the IMUs body frame i the IMU-camera calibration t_{ci}^c, ϕ_c^i is incorporated in the observation model. That is, it has to be combined with the IMUs pose t_{ni}^n, ϕ_n^i to compute the camera pose used for projection, that is:

$$
\begin{aligned}
\mathcal{X}_i &= h^2(X_i, p^2) = R_n^c \cdot (X_i - t_{nc}^n) && (14)\\
&= R_n^c \cdot (X_i - (t_{ni}^n - R_c^n \cdot t_{ci}^c)) && (15)\\
&= {R_c^i}^T \cdot R_n^i \cdot (X_i - t_{ni}^n) + t_{ci}^c && (16)
\end{aligned}
$$

When using stereo cameras, the observation models for the second (slave) camera have to be altered by the master-slave camera translation. That is the stereo baseline has to be subtracted from $\mathcal{X}_i$.

3.2 Filter Composition

For estimating the system state an iterated extended Kalman Filter (IEKF) is used, as described by Petersen[9] et al. It has proven to be able to estimate the IMU-camera calibration consistently and precise. This is due to the fact, that the simple models allow the computation of analytical derivatives. Moreover the prediction is performed with high frequency, resulting in a stable linearization. Finally the update step includes nearly all non linearities of the system to be estimated. As discussed by Lefebvre[12] et al., under such circumstances the IEKF even outperforms the more complex unscented filters.

To keep the rotation parametrization using Euler angles stable even for large rotations, only incremental angles are used. To do so, at time ν an orientation is stored as a pair of a rotation matrix and Euler angles (R^ν, ϕ^ν). Predicting or estimating a change of a coordinate system c^ν to $c^{\nu+1}$ resulting in Euler angles ϕ^ν, the new orientation is stored in the rotation matrix $R^{\nu+1} = R^\nu \cdot R_{(\phi^\nu)}$. The matrix angle pair representing the new orientation is $\left(R^{\nu+1}, 0\right)$. The covariance for the respective Euler angles is modified by linear error propagation. That is, the rows and columns corresponding to ϕ are multiplied with $R_{(\phi^\nu)}{}^T$ and $R_{(\phi^\nu)}$ respectively.

During the first stage, f^1 and h^1 are used alternating to predict the system pose and update the a priori estimate. Therefore the prediction is looped until a marker has been detected for update. This is continued until the standard deviations for ϕ_c^i fall beneath a certain threshold. Following this the second stage is activated and f_2 and h_2 are used to predict/update the state p^2. The threshold is set to 1° since this restricts an update correction to be below 3° at a probability of $> 99.9\%$. Such small changes in the orientation allow for a precise linearization of rotation matrices. This way interferences of non linearities in the rotation alignment estimation with the translational alignment estimation are prevented. Moreover, this level is reached quite fast (see section 4) and provides a reliable initial estimation.

The model switch is done similar to a prediction step. That is, t_{nc} and t_{ci} are summed to determine t_{ni}. According to this ϕ_n^i is computed by ϕ_n^c and ϕ_c^i using the respective external rotation matrices. Finally linear error propagation is applied to determine the covariance matrix for the resulting state p^2. The second stage afterwards performs the final estimation of t_{ci}^c and ϕ_c^i.

Note that the first and second state model include all nuisance and calibration parameters. The first prediction and observation models do not include the accelerations and the translational alignment. This way they are part of the models but are not observable. The second prediction model includes the accelerations and the second observation model the translational alignment. Using the state transfer described above, the correlations between the initial rotation estimates and the second stage rotation and translation estimates are taken into account. By this, the estimation of covariances during Kalman filtering is more stable.

3.3 Application Flow

The calibration software is implemented in a rapid prototyping framework developed in a separate project (see the acknowledgments for project details). This framework makes use of modules grouping associated parts of the application. By this, parts of the software can be easily replaced without recompiling, e.g. for testing different marker detectors or estimators. Moreover it provides an interface for the visualization of processed data. The application is designed to be executed fully automatic and to be usable without parameter tuning. The only exception is, that the noise levels for the IMU (acceleration and turn rate noise) should be known.

The looped work flow starts in stage 1 (first model in section 3.1) with reading and preprocessing the images in parallel for the master and the slave cameras respectively. Following this the markers are detected in both images in parallel. Finally the filter prediction and update steps are applied. As long as the application runs the first stage, the standard deviations σ of the rotational IMU-camera alignment ϕ_c^i are compared to a threshold. This is chosen to be 1°, as described in section 3.2.

When σ is detected to fall below this threshold the first time, the model switch is applied as described in section 3.2. By this the second stage of estimation is activated. In this stage the IMU-camera displacement t_{ci}^c is taken into account in addition to the rotational alignment ϕ_c^i. The complete work flow is visualized in figure 3. The application is terminated when the standard deviations for the IMU-camera calibration fall below the aimed uncertainty or due to a user request.

The application also provides a visualization of the estimation process. It can be used to validate and analyze the estimation results on the fly. This enables the user to adopt the free hand movement to the information gathered so far.

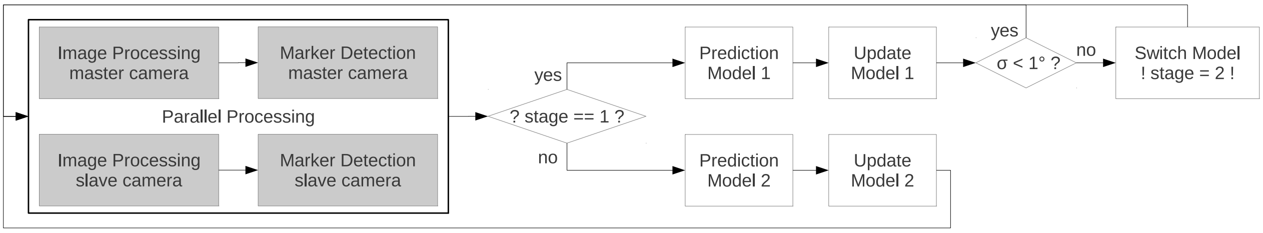
Image Processing master camera
Marker Detection master camera
Parallel Processing
Image Processing slave camera
Marker Detection slave camera
yes
? stage == 1 ?
no
Prediction Model 1
Update Model 1
yes
σ < 1° ?
no
Switch Model ! stage = 2 !
Prediction Model 2
Update Model 2

Figure 3. Flowgraph for module processing within the framework. Grey modules are computed in parallel. “stage” corresponds to the modeling mode discussed in section 3.1. “σ” represents the standard deviations for the IMU-camera rotational alignment.

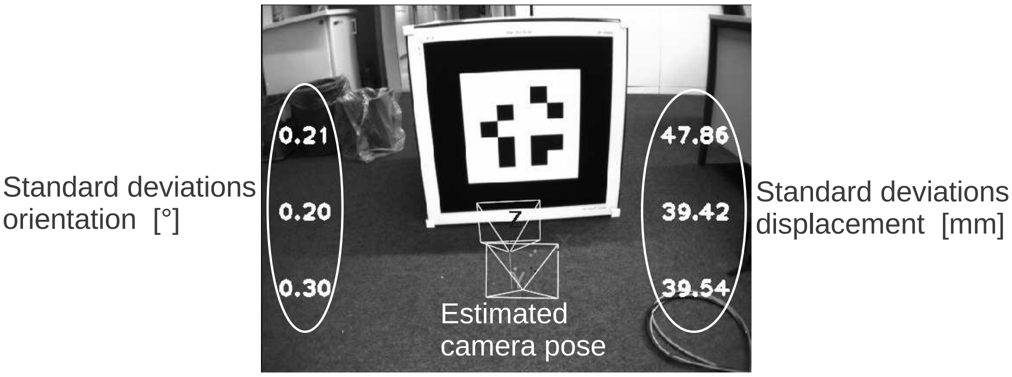
Standard deviations orientation [°]
0.21
0.20
0.30
47.86
39.42
39.54
Standard deviations displacement [mm]
Z
Estimated camera pose

Figure 4. Exemplary visualization for calibration process. On the left: standard deviations for rotational IMU-camera alignment in degrees. On the right: standard deviations for IMU-camera displacement t_{ci}^{c} in millimeters. Pose estimation for IMU is displayed at center.

E.g. consider the standard deviation for the rotational alignment in direction of the X-axis is much higher compared to Y and Z. The user can then be advised to increase the turn rates along the Y- and Z-axis, to improve the estimation. An exemplary process visualization is given in figure 4. For graphical output other resources like debugging information for marker detectors and sensor processing are available also. Moreover timing trees can be displayed allowing for profiling the used software modules.

4. EXPERIMENTS

For the evaluation of calibration quality different tests were carried out. The first group of tests aims at the evaluation of the quality and consistency of estimation. Since ground truth information can hardly be determined, only an empirical evaluation can be done. The results have been compared to manually measured alignments and were found to be plausible. Two validation techniques are especially noticeable. On the one hand, all estimations, done with different alignment and initialization configurations, end up at the same position and orientation in the sensor housing. On the other hand, the estimation of the horizontal alignment has been validated. During all tests, using varying horizontal misalignments, the estimated angles were consistent with the ones, measured using a tubular level (accuracy $< 1^\circ$ according to manufacturer specification). Since the horizontal alignment is highly correlated to the calibration parameters, it can be concluded, that their estimation has comparable precision.

To evaluate the statistical consistency, the most informative system (stereo cameras and 3D-marker) has been used to perform 20 test runs. The calibrations are done for the same alignments using different parameter initializations. Table 1 visualizes the average estimation results and their statistics. As can be seen, the estimated standard deviations for the translational alignment t_{ci}^{c} are the same as the ones empirically determined. From this it can be concluded, that the standard deviations of $\sim 2mm$ reflect the true estimation error. For the rotational alignment ϕ_{c}^{i} the standard deviations are overdetermined but reflect the correct magnitude. A noticeable fact is, that a constant factor of $\sim$5 applies to all three estimated deviations. This implies a systematical error and is subject to further research. The inconsistency for the rotation alignment also causes a reduced consistency in the horizontal alignment ϕ_{e}^{n}.

	t^c_{ci} $[m]$			ϕ^i_c $[°]$			ϕ^n_e $[°]$	
	X	Y	Z	X	Y	Z	X	Z
mean	0.1394	0.0749	0.0116	25.18	-12.79	-29.67	0.263	-0.2963
σ estim.	0.002	0.002	0.0027	0.0642	0.0576	0.0883	0.1007	0.1069
σ empir.	0.0021	0.0018	0.0022	0.276	0.2449	0.4648	0.2265	0.2915

Table 1. Estimation results avg. for 20 tests (45 sec. each) using different initializations. Mean and standard deviations of estimated translational t^c_{ci}, rotational ϕ^i_c and horizontal ϕ^n_e alignment. σ estim.: mean of estimated std devs; σ empir.: empirical std devs of test series.

	t^c_{ci} $[m]$			ϕ^i_c $[°]$		
	X	Y	Z	X	Y	Z
3D-marker stereo	0.1406	0.0731	0.0114	25.08	-12.62	-29.22
3D-marker monocular	0.1422	0.0756	0.0123	25.03	-12.50	-29.57
2D-marker stereo	0.1368	0.0685	0.0148	25.74	-13.05	-28.92
2D-marker monocular	0.1374	0.0724	0.0208	24.76	-12.66	-29.16

Table 2. Comparison of performance for the different hardware setups. For each type 4 tests were carried out with 140 sec. each. All parameter initializations were assumed to be unknown and set to 0.

The second group of tests was performed for different hardware setups. Monocular and stereo systems as well as both marker types (see 2.3) are compared. In table 2 estimation results for all setups are visualized. The estimated values have been averaged over 4 test runs at 140 seconds each. The alignment scenario was the same as for table 1. As can be seen, the use of a marker type has the biggest influence. The estimation stability decreases when using a 2D-marker. Especially the estimation error for the Z-component of the translation t^c_{ci} (cameras view direction) increases by 40% to 90% of the overall length, using table 1 as reference.

The final test group aims at the convergence conditions and stability. In figure 5 an exemplary calibration for the same alignment scenario as discussed above is visualized. The initializations were assumed as unknown and thus set to 0. As can be seen, all estimated parameters converge to the final estimation rapidly. E.g. the rotational X-axis alignment decreases its error from $25.18°$ to $0.4°$ within the first 13.3 seconds (200 images) using table 1 as reference. At time index 6.3 seconds (95 images) the system switched to the second stage. From then on, the translation t^c_{ci} decreases its error from $144mm$ to $2mm$ within 16.7 seconds.

In figure 6 the estimation of horizontal alignment for two calibration runs is visualized. The left graph shows a consistent estimation, as observed for a valid calibration result. During the experiments carried out, some calibration runs produced erroneous output. These outliers are mostly due to accelerations or turn rates, being outside the measurement bounds of the IMU. For those tests the estimation of the horizontal alignment was highly inconsistent, as can be seen in the right graph of figure 6. This allows for an automated alert, when the calibration is expected to be invalid. This aids the user in performing a stable and reliable calibration.

Further testing was applied using larger translational and rotational alignments for unknown initialization. These tests showed, that higher magnitudes of translation do not harm the estimation. While the relative estimation error stays the same, the absolute error increases with increasing translation. This is due to the fact, that the rotational alignment gets more important for large translations. For the same reason that rotation estimation becomes more stable for increasing IMU camera translations. The magnitude of rotational alignment is only bounded by the standard deviations needed for consistent initialization. For standard deviations $> 40°$ parameter updates for the angles of more then $90°$ can occur. This not only harms the linearization of the rotation estimation, but also leads to ambiguities in the Euler representation. This can cause high inconsistencies or even divergence of the filter.

Finally the quality of estimation and convergence speed with respect to the traveled trajectories were analyzed for all used trajectories. For the applied tests, the angular velocity for all axis turned out to have a major influence. This is due to the fact, that rotations in either the camera or the IMU frame cause translations in the other frame. The rotations around the X- and Y-axis are limited due to the cameras narow field of view. This causes a reduced estimation performance in the cameras view direction (Z-axis). Using the 3D-marker this can be compensated, by moving around the marker. This is also visible in the improved estimation of the Z-component of t^c_{ci} for the 3D-marker (see table 2).

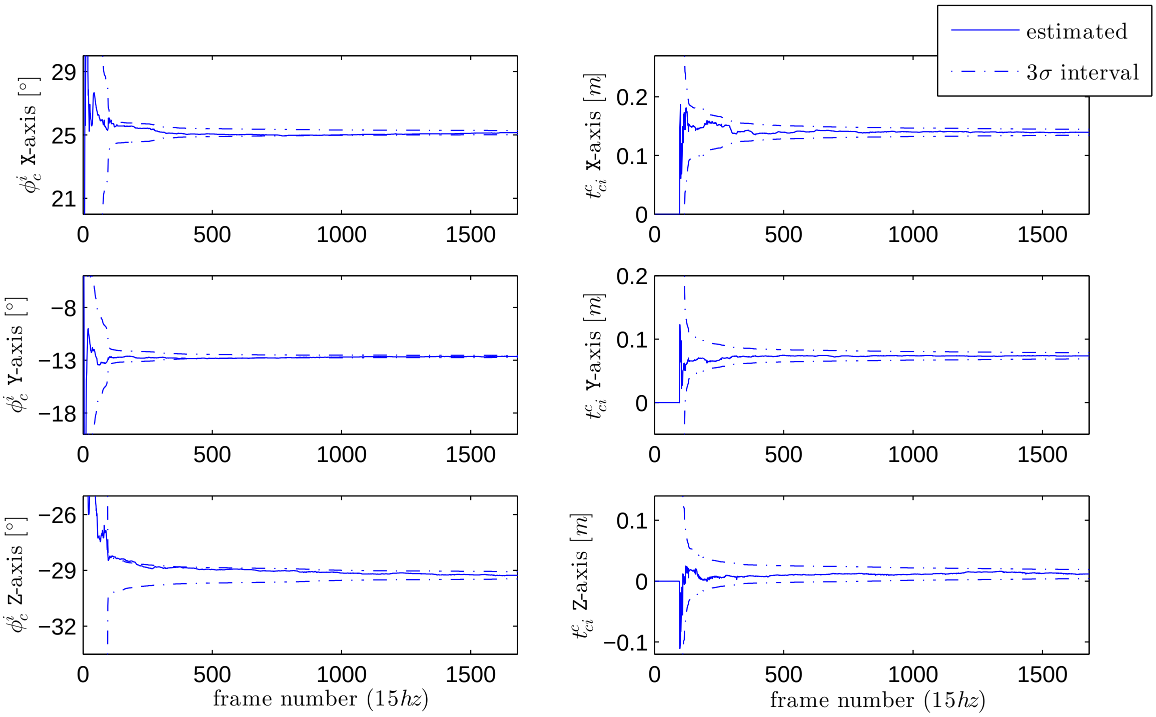

Figure 5. Estimation of rotational ϕ_c^i and translational t_{ci}^c alignment using unknown initialization (ϕ_c^i,t_{ci}^c assumed 0). Dotted lines mark 3σ-confidence interval. Test duration 110 sec.

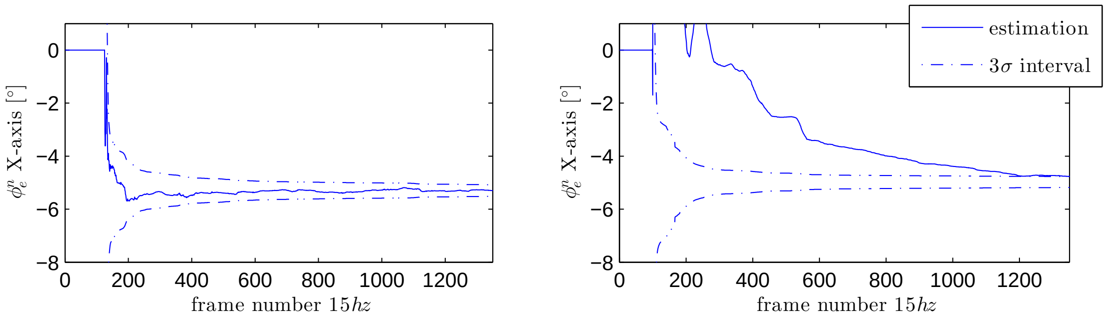

Figure 6. Estimation of horizontal alignment. Left: successful (consistent) estimation; right: erroneous (inconsistent) estimation

All tests were performed in realtime and online. For evaluation purposes estimation results have been logged to text files. The only limiting factor for the systems runtime performance is the cameras framerate, in our case $15hz$. The computational effort for the blocks given in figure 3 is given in table 3.

Module	image preproc.	marker detection	prediction	update	overall
Runtime	5 ms	12 ms	1.5 ms	3.5 ms	22 ms

Table 3. Timings for all modules from figure 3

5. CONCLUSION

We presented an implementation of our novel IMU-camera calibration algorithm. This is capable of estimating the alignment parameters and their covariances in realtime and without parameter tuning. The algorithm is fully automatic and robust to large initialization errors, thanks to a two stage estimation. In contrast to other methods using two stages, the correlations between rotational and translational alignment are taken into account. Moreover, the software is modular, allowing for easily adopting the system to new demands. Finally our experiments showed, that the estimation process is stable and consistent with respect to the different hardware setups and trajectories.

For future research we plan to apply the process noise estimation procedure, proposed by Petersen[13] et al. By this, the noise characteristics of the process model and the IMU hardware can be determined. This increases the quality and consistency of estimation. We are confident, that this approach reduces the rotational consistency problems discussed in section 4. Finally it is planed, to introduce more automated supervising to the software. I.e. the automatic calibration supervision using evaluations of consistency of the horizontal alignment (see section 4).

Acknowledgments

This work has been funded by the “Zukunftsprogramm Schleswig-Holstein (2007-2013)” with funds from the European Commission (EFRE) and Land Schleswig-Holstein, Germany, as part of the Initiative KoSSE, project 122-09-048, and was supported by IBAK Helmut Hunger GmbH & Co. KG.

REFERENCES

[1] Lemaire, T., Berger, C., Jung, I.-K., and Lacroix, S., “Vision-based slam: Stereo and monocular approaches,” *International Journal of Computer Vision* **73**(3), 343–364 (2007).

[2] Hildebrandt, M. and Kirchner, F., “Imu-aided stereo visual odometry for ground-tracking auv applications,” tech. rep., Underwater Robotics Department, DFKI RIC Bremen (2010).

[3] Crowley, J. L., “World modeling and position estimation for a mobile robot using ultrasonic ranging,” in [*IEEE Conference on Robotics and Automation*], 1574–1579 (May 1989).

[4] Ceriani, S., Fontana, G., Giusti, A., Marzorati, D., Matteucci, M., Migliore, D., Rizzi, D., Sorrenti, D. G., and Taddei, P., “Rawseeds ground truth collection systems for indoor self-localization and mapping,” *Autonomous Robots* **27**(4), 353–371 (2009).

[5] Lobo, J. and Dias, J., “Relative pose calibration between visual and inertial sensors.,” *I. J. Robotic Res.* **26**(6), 561–575 (2007).

[6] Mirzaei, F. M. and Roumeliotis, S. I., “A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation,” *IEEE Transaction on Robotics* **24**.

[7] Hol, J., Schön, T., and Gustafsson, F., “Relative pose calibration of a spherical camera and an imu,” in [*Proceedings of the 7th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*], (Sept. 2008).

[8] Kelly, J. and Sukhatme, G. S., “Fast relative pose calibration for visual and inertial sensors,” in [*Experimental Robotics: The Eleventh International Symposium*], Khatib, O., Kumar, V., and Pappas, G. J., eds., *Springer Tracts in Advanced Robotics* **54**, 515–524, Springer, Berlin, Germany (Apr 2009).

[9] Petersen, A. and Koch, R., “A novel state parametrization for stereo-slam,” in [*Proceedings of the International Conference on Computer Vision Theory and Applications*], **2**, 144–153 (February 2012).

[10] Schiller, I., Beder, C., and Koch, R., “Calibration of a pmd camera using a planar calibration object together with a multi-camera setup,” in [*The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*], **Vol. XXXVII. Part B3a**, 297–302 (2008). XXI. ISPRS Congress.

[11] Hartley, R. and Zisserman, A., [*Multiple View Geometry*], Cambirdge University Press, 2 ed. (2003).

[12] Lefebvre, T., Bruyninckx, H., and De Schutter, J., “Kalman filters for non-linear systems: a comparison of performance,” *International Journal of Control* **77**(7) (2004).

[13] Petersen, A. and Koch, R., “Statistical analysis of kalman filters by conversion to gauss-helmert models with applications to process-noise estimation,” in [*Proceedings of ICPR2010*], (August 2010).