Off-the-Shelf Sensor Integration for mono-SLAM on Smart Devices

Philipp Tiefenbacher, Timo Schulze and Gerhard Rigoll
Technische Universität München
philipp.tiefenbacher@tum.de, schulzetimo@gmail.com, rigoll@tum.de

Abstract

This work proposes a fusion of inertial measurement units (IMUs) and a visual tracking system on an embedded device. The sensor-to-sensor calibration and the pose estimation are both achieved through an unscented Kalman filter (UKF). Two approaches for a UKF-based pose estimation are presented: The first uses the estimated pose of the visual SLAM system as measurement input for the UKF; The second modifies the motion model of the visual tracking system. Our results show that IMUs increase tracking accuracy even if the visual SLAM system is untouched, while requiring little computational power. Furthermore, an accelerometer-based map scale estimation is presented and discussed.

1. Introduction

Smart devices are mobile embedded systems equipped with a video camera which enables the integration of simultaneous localization and mapping (SLAM). Mobile augmented reality applications employ the SLAM information. This is of big interest since smart devices are widely spread and augmented reality provides a new way of experiencing content on them. Smart devices, however, include additional sensors (IMUs) which are utilized in this work.

Thus, we focus on the easy integration of inertial measurement units (IMUs) into an existing vision-based SLAM approach without the need of external equipment. We propose both a UKF-based fusion of the sensor data as well as a UKF motion model, which is related to the motion model employed by the parallel tracking and mapping (PTAM) algorithm [8]. The UKF-based data fusion does not interfere with the SLAM system. We apply a nonlinear filter since the accelerometer output affects the device position quadratically.

The PTAM [8] algorithm serves as the SLAM approach of this work since it is fast enough to run on an embedded system and is widely known. The camera and sensors of the consumer device "Microsoft Surface 2 Pro" are used for all our results. The contributions of this work are the following: *a*) Stating a sensor-to-sensor calibration between camera and IMU, *b*) Presenting improved accuracy of velocity-based scale calculation, *c*) Introducing a UKF-based model which incorporates visual SLAM results as measurements, *d*) Comparing the new UKF-based motion model with the one of PTAM .

2. Related Work

Early SLAM methods employed a Kalman filter, whose complexity scales quadratically with map size since each new map point is added to the state vector. More efficient filtering methods such as FastSLAM [10] treat SLAM as a localization problem with a collection of N landmark estimation problems by employing a particle filter. The PTAM [8] algorithm used in this work belongs to the keyframe-based monocular SLAM methods. It differs in a fundamental way from the filtering-based approaches: The knowledge of the system is not represented by a probability distribution but by a subset of images (keyframes) and map points.

Direct visual odometry (VO) methods do not use keypoints but rather include all image information by operating on pixel intensities directly. Dense Tracking and Mapping (DTAM) [11] is a popular algorithm. Its computational complexity, however, is too demanding to be executable on embedded devices.

Today, several works investigated the use of inertial sensors for visual SLAM systems. For example, Servant et al. [15] used inertial sensor data to support a homography-based tracking system. Their extended Kalman filter (EKF) approach indicated that a camera pose estimate based on high-rate sensor data can be particularly useful during fast movements.

Omari et al. [13] reviewed an optical flow-based visual system coupled with inertial measurement units. Their UKF approach converged even in the presence of large initial errors. The employed filter, however, relied only on a single optical flow feature as well as gyroscope and accelerometer measurements. We use the derived pose of a visual SLAM approach as measurement, whereas the IMUs data serves as control input.

Nützi et al. [12] combined an EKF approach with PTAM, a separate camera and inertial sensors. However, they did

not consider the device's orientation in their calculations. In [1], a linear Kalman filter together with a visual system similar to PTAM is fused with visual-inertial data on a tablet device. A linear filter approach, however, cannot describe the quadratic correspondence between accelerometer and device position.

3. The Unscented Kalman Filter (UKF)

The unscented Kalman filter is an extension to nonlinear functions similar to the EKF. The UKF approximation method, however, does not calculate partial derivatives of any form, instead it employs the unscented transformation. The advantages of the UKF compared to the EKF are its second (and higher) order accuracy and its robustness against initial errors [3,5,9]. The UKF introduces a slightly higher computational cost, which is negligible for low dimensional tasks as employed in this work.

3.1. The Unscented Transformation

The unscented transformation is based on the assumption, that it is easier to approximate a Gaussian distribution than an arbitrary non linear function [6].

First, the distribution is sampled through a set of *sigma points* $S = \{\boldsymbol{\mathcal{X}}_0, ..., \boldsymbol{\mathcal{X}}_p\}$. The state means and covariances are $\bar{\boldsymbol{x}}$ and $\boldsymbol{P}_{xx}$. These points are propagated through the nonlinear transformation. The propagated sigma points are used to calculate the new mean measurements $\bar{\boldsymbol{y}}$ and covariance $\boldsymbol{P}_{yy}$

$$\begin{aligned} \boldsymbol{\mathcal{Y}}_i &= f(\boldsymbol{\mathcal{X}}_i, \boldsymbol{u}), \qquad \bar{\boldsymbol{y}} = \sum_{i=0}^{2n} W_i \boldsymbol{\mathcal{Y}}_i, \\ \boldsymbol{P}_{yy} &= \sum_{i=0}^{2n} W_i (\boldsymbol{\mathcal{Y}}_i - \bar{\boldsymbol{y}})(\boldsymbol{\mathcal{Y}}_i - \bar{\boldsymbol{y}})^T, \end{aligned} \tag{1}$$

with $\boldsymbol{u}(t)$ being the control input and W_i is the corresponding weight of $\boldsymbol{\mathcal{X}}_i$ such that $\sum_{i=0}^{2n} W_i = 1$. The basis of the unscented transformation are the sigma points, since they capture the statistical moments of the probability distribution. Several sampling methods for the sigma points have been suggested, which differ in accuracy and computational cost. The next section explains the symmetric sampling method used in this work.

3.2. Symmetric Sigma Point Sampling

The symmetric sampling method [6] uses $p = 2n + 1$ sigma points. They are calculated as

$$\begin{aligned} \boldsymbol{\mathcal{X}}_0 &= \bar{x} & W_0 &= \kappa/(n+\kappa), \\ \boldsymbol{\mathcal{X}}_i &= \bar{x} + (\sqrt{(n+\kappa)\boldsymbol{P}_{xx}})_i & W_i &= 1/(2(n+\kappa)), \\ \boldsymbol{\mathcal{X}}_{i+n} &= \bar{x} - (\sqrt{(n+\kappa)\boldsymbol{P}_{xx}})_i & W_{i+n} &= 1/(2(n+\kappa)), \end{aligned} \tag{2}$$

where n denotes the dimension, $\kappa \in \mathbb{R}$ is for tuning purposes. The "square-root" matrix $\sqrt{\boldsymbol{P}_{xx}}$ is shorthand notation and can be obtained using Cholesky decomposition. The term $(\sqrt{(n+\kappa)\boldsymbol{P}_{xx}})_i$ resembles the i-th row or column of $\sqrt{\boldsymbol{P}_{xx}}$.

4. Camera-IMU Calibration

The relation between camera and IMUs have to be calculated before the IMUs can be used. This section shows the camera-(to)-IMU estimation based on visual and inertial sensor measurements. More precisely, we want the position ${}^I\vec{P}_C$ and orientation ${}^I_C\mathbf{R}$ of the camera relative to the IMU. In the following the rotations are represented by unit quaternions ${}^I_C\mathbf{R} \equiv {}^I_C\hat{q}$.

4.1. Model Description and Observability

Kelly et al. [7] proved that the system described is (locally weakly) observable. The camera pose is the only measurement which is obtained through a known chessboard pattern in the camera-IMU calibration step. Later, it is replaced with the pose calculated by PTAM.

The UKF [3, 9] fuses the visual and inertial measurements. Time synchronization between inertial and visual measurements are implemented such as proposed by Servant et al. [15].

Our state vector has 26 dimensions containing two unit quaternions:

$$\hat{\boldsymbol{x}}(t) = \begin{pmatrix} {}^I_C\hat{q} \\ {}^I\vec{P}_C \\ \vec{b}^g_d \\ \vec{b}^a_d \\ {}^I_W\hat{q}(t) \\ {}^W\vec{P}_I(t) \\ {}^W\vec{V}_I(t) \\ {}^W\vec{g}_0 \end{pmatrix} = \begin{pmatrix} \text{camera-IMU attitude} \\ \text{camera-IMU position} \\ \text{gyroscope bias} \\ \text{accelerometer bias} \\ \text{world-IMU attitude} \\ \text{IMU-world position} \\ \text{IMU-world velocity} \\ \text{gravity} \end{pmatrix}. \tag{3}$$

The IMU signals represent the control input $\boldsymbol{u}(t)$ with $g(t)$ and $a(t)$ being the gyroscope and accelerometer values, respectively. States independent of time t are updated through the *Kalman Gain* (Eq. (17)) solely.

4.2. Process Model

The accelerometer and gyroscope biases are estimated with the IMU-calibration algorithm suggested by Tedaldi et al. [17]. However, we include also the biases in the state vector since we observe a difference between dynamic and static biases. The dynamic biases are introduced due to the IMU calibration failures. The models of the gyroscope and accelerometer are of the same fashion. Only the gyroscope is described in detail:

$$\vec{g}(t) = h(\vec{g}^r, \theta^g) = {}^{BF}_{GF}\boldsymbol{T}\boldsymbol{K}^g(\vec{g}^r(t) + \vec{b}^g + \vec{b}^g_d). \tag{4}$$

${}^{BF}_{GF}\boldsymbol{T}$ is the transformation from the gyroscope to the body frame. $\boldsymbol{K}^g$ holds the estimation of the scaling errors, $\vec{b}^g$ and $\vec{b}^g_d$ are the static and dynamic biases.

The system state evolves in discrete time steps, e.g., for the gyroscope $\vec{r} = \vec{g}(t) \cdot \Delta t$. The attitude of the world relative to the IMU (world-IMU) can then be defined as

$$\delta\hat{q} = \left(\vec{r}/||\vec{r}|| \sin\tfrac{||\vec{r}||}{2}, \cos\tfrac{||\vec{r}||}{2}\right)^T, \tag{5}$$

$${}^I_W\hat{q}(t + \Delta t) = \delta\hat{q}^{-1}(\Delta t) \otimes {}^I_W\hat{q}(t). \tag{6}$$

The velocity and the position of the IMU-world is expressed as

$${}^I\vec{a}(t) = -(\vec{a}(t) - {}^I_W\mathbf{R}(t) \cdot {}^W\vec{g}_0) + {}^I\vec{a}_c(t), \tag{7}$$

with ${}^I_W\mathbf{R}(t) = \mathbf{R}({}^I_W\hat{q}(t))$. $\mathbf{R}(\hat{q})$ is the mapping from any unit quaternion $\hat{q}$ to its rotation matrix $\mathbf{R}$. ${}^I\vec{a}_c(t)$ is the centripetal acceleration

$${}^I\vec{a}_c(t) = \vec{g}(t) \times ({}^I_W\mathbf{R}(t) \cdot {}^W\vec{V}_I(t)), \tag{8}$$

with $\vec{g}(t)$ being the gyroscope's skew-symmetric matrix and $\times$ the cross-product. The velocity and position are calculated via ${}^W\vec{a}(t) = {}^W_I\mathbf{R}(t) \cdot {}^I\vec{a}(t)$ and

$${}^W\vec{V}_I(t + \Delta t) = {}^W\vec{V}_I(t) + {}^W\vec{a}(t) \cdot \Delta t, \tag{9}$$

$${}^W\vec{P}_I(t + \Delta t) = {}^W\vec{P}_I(t) + {}^W\vec{V}_I(t) \cdot \Delta t. \tag{10}$$

4.3. Measurement Model

The camera attitude relative to the world ${}^W_C\hat{q}(t)$ is obtained using the current IMU-world ${}^W_I\hat{q}(t)$ and camera-IMU attitude ${}^I_C\hat{q}$. The camera-world position ${}^W\vec{P}_C(t)$ is calculated in the same fashion, leading to

$$\hat{\mathbf{y}}(t) = \begin{pmatrix} {}^W_C\hat{q}(t) \\ {}^W\vec{P}_C(t) \end{pmatrix} = \begin{pmatrix} {}^W_I\hat{q}(t) \otimes {}^I_C\hat{q} \\ {}^W P_I + {}^W_I\mathbf{R}(t) \cdot {}^I\vec{P}_C \end{pmatrix}. \tag{11}$$

The result of the measurement model is compared to the real measurement in the *innovation* step.

4.4. Filtering Steps

The following filtering steps are conducted in each discrete time step:

- The IMU variances $\boldsymbol{Q}_k$ are added to the state covariance $\boldsymbol{P}_{xx}$. Then, the sigma points are created from $\boldsymbol{P}_{xx}$ by applying the symmetric sampling method. The involvement of the variances $\boldsymbol{Q}_k$ allows the filter to model noise.
- The sigma points are propagated through the nonlinear process model of Section 4.2:

$$\boldsymbol{\mathcal{X}}_{i,k+1} = f(\boldsymbol{\mathcal{X}}_{i,k}, \boldsymbol{u}_k). \tag{12}$$

- The a priori estimate of the state is approximated by its weighted mean

$$\hat{\boldsymbol{x}}^-_{k+1} = \sum_{i=0}^{p} W_i \boldsymbol{\mathcal{X}}_{i,k+1}, \tag{13}$$

and the a priori state covariance $\boldsymbol{P}^-_{xx}$ as

$$\boldsymbol{P}^-_{xx} = \sum_{i=0}^{p} W_i (\boldsymbol{\mathcal{X}}_{i,k+1} - \hat{\boldsymbol{x}}^-_{k+1})(\boldsymbol{\mathcal{X}}_{i,k+1} - \hat{\boldsymbol{x}}^-_{k+1})^T. \tag{14}$$

- Then, the processed sigma points are propagated through the measurement model of Section 4.3 ($\boldsymbol{\mathcal{Y}}_{i,k+1} = h(\boldsymbol{\mathcal{X}}_{i,k+1})$) the same way as in the process model. Consequently, the a priori measurement covariance is

$$\boldsymbol{P}^-_{yy} = \sum_{i=0}^{p} W_i (\boldsymbol{\mathcal{Y}}_{i,k+1} - \hat{\boldsymbol{y}}^-_{k+1})(\boldsymbol{\mathcal{Y}}_{i,k+1} - \hat{\boldsymbol{y}}^-_{k+1})^T. \tag{15}$$

$\hat{\boldsymbol{y}}^-_{k+1}$ is the weighted a priori measurement mean similar to Equation (13).

- Using the innovation covariance $\boldsymbol{P}_{vv} = \boldsymbol{P}^-_{yy} + \boldsymbol{R}_{k+1}$ the *Kalman Gain* is retrieved with $\boldsymbol{\mathcal{K}} = \boldsymbol{P}_{xy}(\boldsymbol{P}_{vv})^{-1}$, whereas the cross correlation matrix $\boldsymbol{P}_{xy}$ is

$$\boldsymbol{P}_{xy} = \sum_{i=0}^{p} W_i (\boldsymbol{\mathcal{X}}_{i,k+1} - \hat{\boldsymbol{x}}^-_{k+1})(\boldsymbol{\mathcal{Y}}_{i,k+1} - \hat{\boldsymbol{y}}^-_{k+1})^T. \tag{16}$$

- Update of the a posteriori state and covariance

$$\hat{\boldsymbol{x}}^+_{k+1} = \hat{\boldsymbol{x}}^-_{k+1} + \boldsymbol{\mathcal{K}}\boldsymbol{\nu} \tag{17}$$

$$\boldsymbol{P}^+_{xx} = \boldsymbol{P}^-_{xx} - \boldsymbol{\mathcal{K}}\boldsymbol{P}_{vv}\boldsymbol{\mathcal{K}}^T \tag{18}$$

with $\boldsymbol{\nu} = \boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}^-_{k+1}$, which is called *innovation*. $\boldsymbol{y}_{k+1}$ holds the real measurement obtained through the chessboard pattern or PTAM.

5. IMU-based Tracking

We propose two ways for including the IMUs. First, PTAM is kept untouched and the final camera pose of PTAM serves as measurement for the UKF (*pure UKF*). Second, the PTAM algorithm consists of a motion model for a prior estimate of the pose. This PTAM motion model (PMM) can be replaced with the a priori state (Eq. (13)) of the UKF. We call the UKF motion model UMM.

5.1. PTAM Motion Model (PMM)

The motion model consists of a decaying velocity model for the prior position estimate. The innovations for position, angular and linear velocity are obtained from the Special-Euclidean-Group-3 (SE3) difference between successive frames. More precisely, using exponential coordinates

$$\mathbf{P}_i = \exp(\vec{r}_i) \cdot \mathbf{P}_{i-1} \tag{19}$$

with decaying linear and angular velocity

$$\vec{\tau}_i = 0.9 \cdot (0.5 \cdot \vec{\tau}_{i-1} + 0.5 \cdot \ln(\mathbf{P}_{i-1} \cdot \mathbf{P}_{i-2}^{-1})). \quad (20)$$

Although this model already estimates the angular velocity, we incorporate a separate template-based tracking procedure suggested by Benhimane et al. [2]. This approach is more accurate but also computationally more complex due to the minimization of a sum-of-squared-difference between templates.

5.2. Pure UKF & UKF Motion Model (UMM)

Our *pure UKF* approach utilizes only the PTAM output as measurement for the UKF as proposed in Section 4.3. This way, the UKF does not interfere with the PTAM algorithm. The UMM approach consists of exactly the same steps, and additionally substitutes the PTAM motion model with the a priori pose estimate of the UKF.

The a posteriori state (Eq. (17)) is calculated after each iteration of the PTAM. This estimate is the fusion of the measurement, the PTAM-derived camera pose, and the a priori state $\hat{\boldsymbol{x}}_{k+1}^{-}$. The result of the UKF a posteriori state $\hat{\boldsymbol{x}}_{k+1}^{+}$ is used as final pose.

The static and dynamic biases are considered in Equation (4). The state vector is similar to Equation (3) and has the form

$$\hat{\boldsymbol{x}}(t) = \left({}_{W}^{I}\hat{q}(t), {}^{W}\vec{P}_I(t), {}^{W}\vec{V}_I(t), s_v, {}^{W}\vec{G}\right)^T, \quad (21)$$

with s_v being the velocity scale. The camera attitude ${}_{C}^{I}\hat{q}$ and position ${}^{I}\vec{P}_C$ are estimated in Section 4. The states are left out since the relation is static. The resulting vector $\hat{\boldsymbol{x}}(t)$ has only 14 dimensions. The estimation of the map scale is an important difference to the sensor-to-sensor calibration. This is necessary due to the integration of inertial translational data. The scale factor s_v is applied to the velocity change, yielding to a modification of Equation (8)

$${}^{I}\vec{a}_c(t) = \vec{g}(t) \times ({}_{W}^{I}\mathbf{R}(t) \cdot {}^{W}\vec{V}_I(t)/s_v), \quad (22)$$

$${}^{W}\vec{V}_I(t + \Delta t) = {}^{W}\vec{V}_I(t) + s_v \cdot {}^{W}\vec{a}(t) \cdot \Delta t. \quad (23)$$

Ideally, the velocity scale s_v converges to the true scale. This is, however, only partly the case as will be shown in Section 6.1.

6. Evaluation

A professional, external tracking system by *ART* provides the ground truth of our comparisons. The tracking system includes five high-resolution 60 Hz cameras in combination with passive markers. The markers are mounted on the mobile device. We performed a hand-eye calibration between marker- and camera-center. The data is recorded and evaluated offline. Figure 1 illustrates the texture-rich setting. All sets are recorded with 20 FPS, 640x480p resolution, 60 Hz IMU-readings and a duration of about $100-120$ seconds per set.

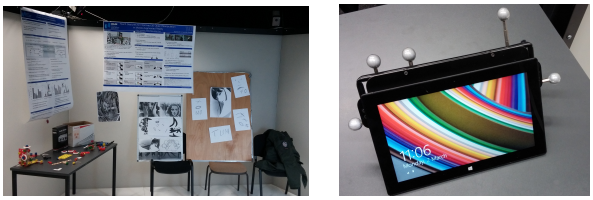

Figure 1: Tracking environment and mobile device with passive markers.

6.1. Scale Estimation

The pose of the camera is known as its attitude and position. The former is scale-invariant and can be used directly with real world data, the latter is not. The scale s can be described as the relation between accelerometer and position-related visual data. There are two ways of including it in the filtering process.

The professional tracking system defines the true map scale s_m. The scale s can either be computed in conjunction with the velocity s_v, resulting in Equation (22) and (23) or with the position s_p

$${}^{W}\vec{P}_I(t + \Delta t) = {}^{W}\vec{P}_I(t) + s_p \cdot {}^{W}\vec{V}_I(t) \cdot \Delta t. \quad (24)$$

6.1.1 Scale Sets

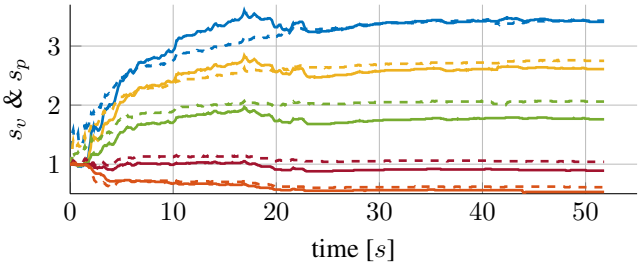

Figure 2: Progress of the velocity s_v and position scale s_p (dashed lines) for the same recording and different true map scales s_m.

The test set consists of five recordings. We choose solely sets without loss of tracking. The scale of the PTAM map can be modified by changing the distance between the first two keyframes used in the initialization process.

The data of the sets are multiplied with a factor to modify the true map scales s_m and generalize our results. A uniform distribution between 1 and 5 determines this factor. The factor of the initial UKF-scale is set to 1. The resulting estimation process for one recording is displayed in Figure 2. It is visible that the scale converges to the final value after $20-25$ s regardless of the map scale size.

6.1.2 Scale Results

Both scale estimation methods run on the same data but separately. The result of each run and the best fit are displayed in Figure 3. Noticeable are the linear dependencies of both scale types. The relative error between the true map scale s_m and the linear fit are 0.11 ± 0.14 and 0.08 ± 0.08 for s_p and s_v, respectively. The velocity scale s_v delivers more accurate estimates than the position applied scale s_p. The IMU velocity (Eq. (23)) is close to the pure accelerometer data as such it permits a more reproducible feedback and faster convergence compared to the IMU position ${}^W\vec{P}_I(t)$. The works [1,12,16] all include scale estimation but did not publish data for comparison.

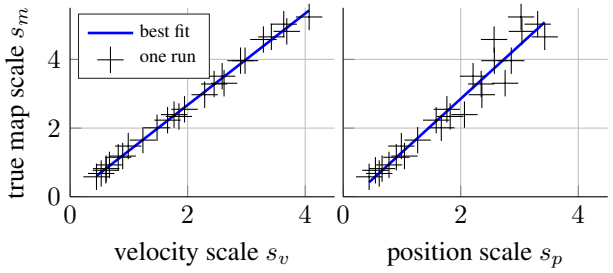

Figure 3: Linear dependency between UKF-estimated scales and the true map scale. Velocity and position scale have the same ground truth.

6.2. Motion Model Comparison

Motion models predict prior poses for trackers. The precision and stability of these motion models have a direct influence on tracking performance. This section compares the fully inertial sensor-based motion model UMM with the original PMM. We stop the visual update for $10\,s$ for that purpose. The device is moved around. Both motion models apply their a priori estimate as measurement during this time. Figure 4 illustrates the mean error relative to the ground truth for 10 different recordings.

The attitude error in Figure 4a shows the advantages of a gyroscope-based attitude update. The PMM converges quickly to zero change, whereas the UMM keeps track of the device's attitude.

This is not the case for the translational error. Even a small constant attitude error of 1° leads to a faulty acceleration of around $0.17 m/s^2$. This results in huge position errors after a few seconds as depicted in Figure 4b.

A way to handle this issue is to apply confidence intervals on the difference between observed and predicted measurements [4]. We do not tackle this problem yet.

Instead, we remove the translational part of the UMM and incorporate the translation of the PMM in a *PUMM* version of the motion model.

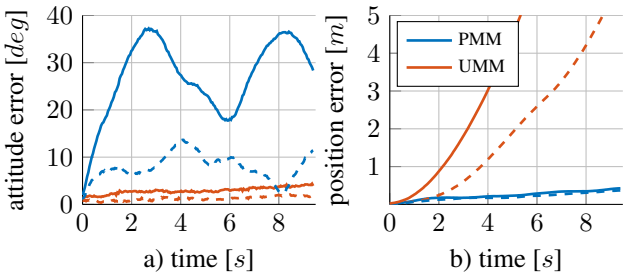

Figure 4: Error-time dependency for pose estimation without visual update. Dashed lines indicate the standard deviation.

6.3. Pose Accuracy

The *PUMM* uses PMM and UMM in parallel threads without interference. Table 1 displays the results of the pose accuracy of five recordings without tracking failure. *PUMM* modifies the PTAM motion model and sets the PTAM pose as final result. The $PUMM_{UKF}$ uses the PTAM pose as measurement as with the *pure UKF*.

PUMM cannot increase accuracy since the position estimate worsens (-3.8%) by improving the attitude proportionally (3.1%). $PUMM_{UKF}$ performs better than both PTAM-based final poses. The average errors of position ($\varnothing\epsilon_{pos}$) and attitude ($\varnothing\epsilon_{att}$) are reduced in comparison to PTAM by 12.7% and 6.0%, respectively. Keeping the PTAM untouched and using its result as measurement for our proposed UKF (*pure UKF*) leads to 18.4% and 5.2% improved accuracy.

method	$\varnothing\epsilon_{pos} \pm \sigma\,[m]$	$\varnothing\epsilon_{att} \pm \sigma\,[deg]$
PTAM	0.158 ± 0.086	2.611 ± 0.969
PUMM	0.164 ± 0.089	2.529 ± 0.921
$PUMM_{UKF}$	0.138 ± 0.075	2.454 ± 0.894
pure UKF	0.129 ± 0.072	2.475 ± 0.889

Table 1: Averaged errors of position and attitude for different methods. Final poses above the middle line rely on the PTAM, below on UKF a posteriori estimate.

6.4. Computational Costs

The runtime of the system is computed based on 2000 frames with the *pure UKF* method. The final map consists of 18 keyframes and 1700 map points. TrackMap denotes the search for map points and subsequent pose update procedures. FAST denotes the detection of FAST corners [14]. The sigma points are propagated on average 3.82 times per image. The Cholesky decomposition is executed as often as there are frames in the set. Table 2 depicts the contributions of the components to the total computation time.

The UKF is cheaper than the PMM calculation. The most time-consuming part of the UKF is the sigma point propagation (3.10%). The Cholesky decomposition con-

tributes only a small fraction to the total costs (0.20%). The decaying velocity model of the PMM comes for free and the $6.10\,\%$ might be saved by employing the template tracking only when necessary.

TrackMap	$71.93\,\%$	PMM	$6.10\,\%$
FAST	$17.65\,\%$	UKF	$4.32\,\%$

Table 2: Relative computational costs of the algorithm components.

7. Conclusion

We stated an unscented Kalman filter approach for the fusion of embedded IMUs with visual data. The proposed approach enables pose tracking as well as camera-IMU calibration. We presented two methods in the context of pose tracking. The first incorporates the a priori state of the UKF with a motion model of a vision-based tracker. In the second method, the pose obtained through visual data serves only as measurement for the UKF. The a posteriori state of the UKF defines the pose.

Our results disclose that the motion model is rather unstable since the accelerometer corrupts the prior position estimates of the SLAM system. This is due to its quadratic propagation of errors. Thus, minimal noise already affects the device position. The gyroscope, however, delivers precise attitudes even in case of camera failure. It increases attitude's accuracy in comparison to the PMM. Consequently, we recommend to integrate only the gyroscope into an IMU-based motion model.

The second method demonstrates an easy and computationally cheap way to improve tracking accuracy with the help of IMUs. The accuracy increases by about 18% and 5% in position and attitude, respectively, without the need of modifying the existing visual tracker. This improvement raises the computational complexity by just 4.32%.

Lastly, we showed that the velocity estimate of an accelerometer is more reliable for the calculation of the map scale than the derived position. This way, a scale error within a margin of 8% is feasible.

We publish the code of both markerless IMU-based tracking and sensor-to-sensor calibration at `www.mmk.ei.tum.de/sensorintegrationslam`.

References

[1] Y. Aksoy and A. A. Alatan. Uncertainty Modeling for Efficient Visual Odometry via Inertial Sensors on Mobile Devices. In *Proceedings of International Conference on Image Processing*. IEEE, 2014.

[2] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 1, pages 943–948. IEEE, 2004.

[3] J. L. Crassidis and F. L. Markley. Unscented Filtering for Spacecraft Attitude Estimation. *Journal of Guidance, Control, and Dynamics*, 26(4):536–542, 2003.

[4] J. D. Hol. Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS. *Linköping studies in science and technology*, (1368), 2011.

[5] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *Transactions on Automatic Control*, 45(3):477–482, 2000.

[6] S. J. Julier and J. K. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. In *Proceedings of Signal Processing, Sensor Fusion, and Target Recognition*, volume 3068, pages 182–193, 1997.

[7] J. Kelly and G. S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *International Journal of Robotics Research*, pages 56–79, 2011.

[8] G. Klein and D. Murray. Improving the agility of keyframe-based SLAM. In *Proceedings of the European Conference on Computer Vision*, pages 802–815. Springer, 2008.

[9] E. Kraft. A Quaternion-based Unscented Kalman Filter for Orientation Tracking. In *Proceedings of the International Conference of Information Fusion*, volume 1, pages 47–54. IEEE, 2003.

[10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. In *Proceedings of the International Conference on Artificial Intelligence*, pages 1151–1156. IEEE, 2003.

[11] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *Proceedings of the International Conference on Computer Vision*, pages 2320–2327. IEEE, 2011.

[12] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM. *Journal of Intelligent & Robotic Systems*, pages 287–299, 2011.

[13] S. Omari and G. Ducard. Metric visual-inertial navigation system using single optical flow feature. In *Proceedings of the European Control Conference*, pages 1310–1316. Springer, 2013.

[14] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Proceedings of the International Conference on Artificial Intelligence and Computer Vision*, volume 2, pages 1508–1515. IEEE, 2005.

[15] F. Servant, P. Houlier, and E. Marchand. Improving monocular plane-based SLAM with inertial measures. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 3810–3815. IEEE, 2010.

[16] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live Metric 3D Reconstruction on Mobile Phones. In *Proceedings of International Conference on Computer Vision*, pages 65–72. IEEE, 2013.

[17] D. Tedaldi, A. Pretto, and E. Menegatti. A robust and easy to implement method for IMU calibration without external equipments. In *Proceedings of International Conference on Robotics and Automation*, pages 3042–3049. IEEE, 2014.