Using the Marginalised Particle Filter for Real-Time Visual-Inertial Sensor Fusion

Gabriele Bleser* Didier Stricker†

Department for Virtual and Augmented Reality
Fraunhofer IGD

ABSTRACT

The use of a particle filter (PF) for camera pose estimation is an ongoing topic in the robotics and computer vision community, especially since the FastSLAM algorithm has been utilised for simultaneous localisation and mapping (SLAM) applications with a single camera. The major problem in this context consists in the poor proposal distribution of the camera pose particles obtained from the weak motion model of a camera moved freely in 3D space. While the FastSLAM 2.0 extension is one possibility to improve the proposal distribution, this paper addresses the question of how to use measurements from low-cost inertial sensors (gyroscopes and accelerometers) to compensate for the missing control information. However, the integration of inertial data requires the additional estimation of sensor biases, velocities and potentially accelerations, resulting in a state dimension, which is not manageable by a standard PF. Therefore, the contribution of this paper consists in developing a real-time capable sensor fusion strategy based upon the marginalised particle filter (MPF) framework. The performance of the proposed strategy is evaluated in combination with a marker-based tracking system and results from a comparison with previous visual-inertial fusion strategies based upon the extended Kalman filter (EKF), the standard PF and the MPF are presented.

Keywords: real-time, sensor fusion, inertial sensors, nonlinear filtering, (marginalised) particle filter, (extended) Kalman filter, FastSLAM

Index Terms: G.3 [Probability and statistics]: Experimental design, Markov processes, Multivariate statistics, Nonparametric statistics, Probabilistic algorithms (including Monte Carlo), Statistical computing; I.2.9 [Artificial intelligence]: Robotics—Kinematics and dynamics, Sensors; I.2.10 [Artificial intelligence]: Vision and Scene Understanding—Motion, Video analysis; I.4.8 [Image processing and computer vision]: Scene analysis—Motion, Sensor fusion, Tracking; I.3.m [Computer graphics]: Miscellaneous—Augmented Reality; General Terms: Algorithms, Design, Experimentation, Performance, Theory, Verification

1 INTRODUCTION

Solving the camera pose estimation problem with a particle filter (PF) is an ongoing trend in the computer vision community, especially in the context of simultaneous localisation and mapping (SLAM) applications based on the FastSLAM algorithm. FastSLAM was originally developed in the robotics community as a scalable method for localising a vehicle navigating with three degrees of freedom (DOF) in an unkown environment [1, 2] and has recently been transferred to six DOF single-camera SLAM [3, 4]. Besides the significant increase in the degrees of freedom, one major problem in taking this step lies in the poor proposal distribution[1] of the camera pose particles obtained due to missing control information about the motion of a handheld camera. FastSLAM 2.0 compensates this problem by utilising the latest observation to guide the particles into regions of high likelihood [6] and has been applied in [4]. Another work presented in [7] applies particle annealing [8]. The particles are concentrated to important areas of the state space by performing several PF iterations for each timestep while adapting the perturbation and measurement noise settings. A third way of improving the proposal distribution is to make assumptions about the camera motion in specific situations. E.g. in [9], where close range, desk based camera localisation is considered, the camera motion is assumed to be orbital.

The solution proposed in this paper is to compensate the missing control information with measurements from low-cost inertial sensors (gyroscopes and accelerometers). The fusion of vision-based and inertial tracking technology has been proposed several times in the past but mostly based upon the extended Kalman filter (EKF) as estimation tool [10–16]. Using the PF as fusion filter has been hardly considered. [17] describes an ad hoc method for incorporating gyroscope measurements by sampling the rotation angles according to the measured angular velocities while applying a random walk to the position. In [18] both types of inertial data, angular velocities and linear accelerations, are used and the state vector includes velocities and linear accelerations. However, the measurement models are not described in detail. In the SLAM system presented by Schön et al in [19,20] the state space is further augmented with time-varying sensor biases resulting in a high state dimension, which is not manageable by a standard PF. As the velocities, linear accelerations and biases are conditionally linear and subject to Gaussian noise, the marginalised particle filter (MPF) is applied — that is these states are marginalised and estimated in separate per-particle Kalman filters (KF) — in order to obtain better estimates. The state-space model is the same as in [15, 16] but reformulated to fit the MPF framework as introduced in [21]. However, the computational complexity of the resulting algorithm is inevitably high and can therefore not be used in a real-time application.

This paper is inspired by the system above. The MPF framework is used as a basis for fusing angular velocities, linear accelerations and 2D/3D point correspondences from the image analysis to a camera pose estimate. The gyroscope biases are also estimated online in order to stabilise and improve the tracking results. However, the contribution of this paper is to develop a fusion strategy, which is able to operate robustly in real-time. This is achieved in two major steps: 1) the design of a reduced order state-space model and 2) the extension of the MPF algorithm as presented in [21] with concepts like automatic model-switching, improved orientation proposals, adaptive process noise and mixture proposals for the translational states.

This paper focuses on the question of how to incorporate inertial data in real-time into the camera pose estimation based on the

*e-mail: gbleser@igd.fhg.de
†e-mail: didier.stricker@igd.fhg.de

[1] The proposal distribution (also importance density) is the distribution used to spread the particles [5]. In this paper the proposal is chosen to be the prior density.

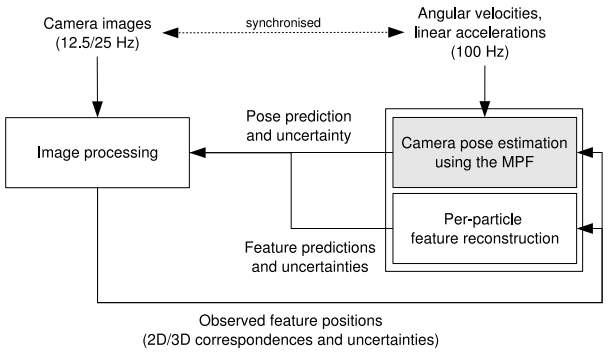


Figure 1: The interface of the proposed visual-inertial pose estimation method (marked grey) and how it can be embedded into a complete visual-inertial SLAM system: assuming that a multi-rate, synchronised stream of inertial readings and camera images is passed to the system, the inertial readings up to the time of the current image are batch processed in the MPF providing a pose for predicting the feature positions. The registered positions are in return used to reweight the camera pose particles during the MPF measurement update and afterwards, in the SLAM case, to update the 3D positions of the registered features in each particle.

MPF. The aim is to provide a pose estimation algorithm that is compatible to FastSLAM and can be used, if inertial sensors are available. Markerless feature tracking and feature reconstruction are not addressed here. These issues have been investigated in previous works [22, 23] and potential solutions are ready to be integrated with this contribution. However, in order to provide a bigger context for the work presented here, Figure 1 outlines not only the interface to the proposed pose estimation method but also how it can be embedded into a complete visual-inertial SLAM system.

The paper is organised as follows. Section 2 introduces some basic facts about the MPF providing the theoretical background for the paper. Section 3 presents the proposed real-time capable visual-inertial sensor fusion strategy. Evaluation results from a comparison of this method with the previous visual-inertial fusion strategies mentioned above are presented in Section 4. A marker based tracking system has been used during the experiments to produce the vision measurements. Section 5 draws final conclusions.

2 Theoretical background of the MPF

For the reader's convenience the MPF is briefly introduced here. A detailed derivation can be found in [21] and the references therein.

The MPF is a combination of the standard PF [5, 24] and the KF [25] where conditionally linear sub-structures subject to Gaussian noise are marginalised and estimated in separate per-particle KFs. Nonlinear states are marginalised using EKFs. The motivation to use the MPF is to obtain better estimates, i.e. estimates with reduced variance and/or reduced computational costs compared to the standard PF (cf. [26, 27]).

The MPF applies to mixed linear/nonlinear state-space models in the general form[2]:

$$\mathbf{x}^n_{t+T} = f^n_t(\mathbf{x}^n_t, \mathbf{u}^n_{t+T}) + A^n_t(\mathbf{x}^n_t)\mathbf{x}^l_t + G^n_t(\mathbf{x}^n_t)\mathbf{v}^n_t \tag{1a}$$

$$\mathbf{x}^l_{t+T} = f^l_t(\mathbf{x}^n_t) + A^l_t(\mathbf{x}^n_t)\mathbf{x}^l_t + B^l_t(\mathbf{x}^n_t)\mathbf{u}^l_{t+T} + G^l_t(\mathbf{x}^n_t)\mathbf{v}^l_t \tag{1b}$$

$$\mathbf{y}_t = h_t(\mathbf{x}^n_t) + C_t(\mathbf{x}^n_t)\mathbf{x}^l_t + \mathbf{e}_t \tag{1c}$$

with state vector $\mathbf{x}_t$, control signals $\mathbf{u}_t$ and measurements $\mathbf{y}_t$ with process noise $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, Q_t)$ and measurement noise $\mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, R_t)$. The state vector is partitioned into a nonlinear part $\mathbf{x}^n_t$ and a linear part $\mathbf{x}^l_t \sim \mathcal{N}(\hat{\mathbf{x}}^l_t, P_t)$ resulting in separate time update equations for the PF (1a) and the KFs (1b). The control signals and the process noise are also given separately for each state partition. A_t, B_t, C_t and G_t are matrices depending on $\mathbf{x}^n_t$ and f_t and h_t denote nonlinear functions.

The above equations show the most general form of the mixed linear/nonlinear state-space model. Many applications allow for special cases with relaxed assumptions and possibly reduced computational demand (cf. [21]). For instance, in the case of linear system dynamics and nonlinear measurements one covariance matrix can be used for all particles. Unfortunately this does not apply to the application considered here. The original MPF method is given in Algorithm 1. A Gaussian state estimate is obtained from the MPF by computing the weighted mean and covariance of the particle distribution. The equations are given in for instance [28].

Algorithm 1: General MPF

Let $\langle \mathbf{x}^n_{t|t-T}, \hat{\mathbf{x}}^l_{t|t-T}, P_{t|t-T}, w_{t-T}\rangle^{[i]}$, $i = 1, \ldots, N$ be the set of weighted particles representing the prior probability distribution at time t. The recursive state estimation consists of the following steps:

1. PF MU[a]: compute the importance weights $w^{[i]}_t \propto p(\mathbf{y}_t|\mathbf{x}^{n,[i]}_{0:t}, \mathbf{u}^n_{0:t}, \mathbf{y}_{0:t-T})$ based on (1c) and normalise.
2. Resample: draw particle $[i]$ with probability $w^{[i]}_t$.
3. KF MU: Compute $p(\mathbf{x}^{l,[i]}_t|\mathbf{x}^{n,[i]}_{0:t}, \mathbf{u}^l_{0:t}, \mathbf{y}_{0:t})$ based on (1c).
4. PF TU[b]: sample $\mathbf{x}^{n,[i]}_{t+T|t} \sim p(\mathbf{x}^n_{t+T}|\mathbf{x}^{n,[i]}_{0:t}, \mathbf{u}^n_{0:t+T}, \mathbf{y}_{0:t})$ based on (1a).
5. KF CMU[c]: compute $p(\mathbf{x}^{l,[i]}_t|\mathbf{x}^{n,[i]}_{0:t+T}, \mathbf{u}^l_{0:t}, \mathbf{y}_{0:t})$ based on (1a).
6. KF TU: compute $p(\mathbf{x}^{l,[i]}_{t+T}|\mathbf{x}^{n,[i]}_{0:t+T}, \mathbf{u}^l_{0:t+T}, \mathbf{y}_{0:t})$ based on (1b).

[a] measurement update
[b] time update
[c] corrective measurement update

3 Using the MPF for real-time visual-inertial sensor fusion

A novel fusion strategy — capable of estimating the camera pose from angular velocities, linear accelerations and 2D/3D point correspondences in real-time — is now developed within the MPF framework as introduced above. Section 3.1 introduces the notation used subsequently, Section 3.2 describes the design of the mixed linear/nonlinear state-space model and Section 3.3 applies the MPF to the considered application and presents extensions made to the general MPF algorithm.

3.1 Notation

The following coordinate systems are used: the world frame, w, (fixed to the target scene model), the camera frame, c, (fixed to the moving camera), the sensor frame, s, (fixed to the moving inertial measurement unit (IMU)) and the normalised image frame, n, (fixed to the camera images with focal length $f = 1$). The reference frame, in which a quantity is resolved, is indicated by sub-

[2] The model given in [21] has been extended with control inputs here.

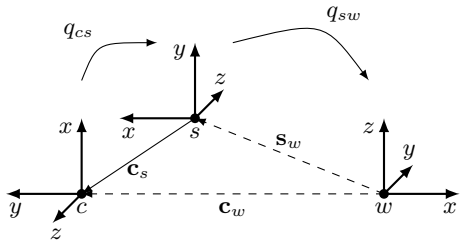


Figure 2: Illustration of the different 3D coordinate systems and how they are related. Rigid transformations are indicated by solid lines, non-rigid by dashed lines.

scribing the accordant abbreviation. The abbreviation is also used for indicating the origin of a reference frame, e.g. $\mathbf{s}_w$ is the origin of the IMU s, given in the world frame w. $\dot{\mathbf{s}}_w$ and $\ddot{\mathbf{s}}_w$ denote the velocities and accelerations, respectively, of s. Transformation subscripts contain two letters denoting the mapping. Unit quaternions are used to parametrise rotations, for instance $\mathbf{q}_{sw}$ describes the rotation from the world frame, w, to the IMU frame, s. The corresponding rotation matrix is denoted Q_{sw}. The quaternion product is denoted $\odot$. See [29] for more information on quaternions and the conversion formulas. Figure 2 illustrates the 3D coordinate systems and transformations used throughout this paper.

$\mathcal{N}(\hat{\mathbf{x}}, P)$ denotes a multi-dimensional normal distribution with mean $\hat{\mathbf{x}}$, covariance P and corresponding Gaussian probability density function (pdf) $\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, P)$ in $\mathbf{x}$.

3.2 Designing the state-space model

To obtain a real-time capable filter algorithm, the order of the state space has to be reduced. This can be achieved by modelling the inertial readings as control inputs (*reduced order model*) instead of measurements (*complex model*). Both approaches have been proposed in the past [17, 19], however, mostly using the EKF for filtering [14–16, 30]. By treating the angular velocities and accelerations as input signals to the dynamic model, six states and two measurement update steps can be saved[3]. A comparison of both approaches using the EKF implementation and experimental setup of [16] showed that the reduced order model performs as well as the complex model with reduced computational costs. Another experience from these preliminary experiments, which due to the limited space are not presented here, concerns the estimation of sensor biases. The gyroscope biases converged reliably while the accelerometer biases were hardly observable among other errors such as calibration and model errors. These tended to show up incorrectly in the bias parameters causing instabilities rather than improving the overall tracking performance significantly. These observations and the fact that the gyroscope biases are the dominant sources of error, not only for the rotational but — if accelerometers are used for estimating the body acceleration — also for the translational states[4], led to the decision to estimate only the gyroscope biases online.

The reduced order state vector $\mathbf{x}$ comprises the position $\mathbf{s}_w$, the velocities $\dot{\mathbf{s}}_w$ and the orientation $\mathbf{q}_{sw}$ of the IMU and the gyroscope biases $\mathbf{b}_s^\omega$. With the measured angular velocities and accelerations as control inputs $\mathbf{u}^T = [\mathbf{y}_{s,t}^{\omega T}\ \mathbf{y}_{s,t}^{aT}]$, the dynamic (constant acceleration, constant angular velocity) model is:

$$\begin{bmatrix} \mathbf{s}_{w,t+T} \\ \dot{\mathbf{s}}_{w,t+T} \\ \mathbf{q}_{sw,t+T} \\ \mathbf{b}^\omega_{s,t+T} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{w,t} + T\dot{\mathbf{s}}_{w,t} + \frac{T^2}{2} Q_{ws,t}(\mathbf{y}^a_{s,t+T} - \mathbf{v}^a_{s,t}) + \frac{T^2}{2}\mathbf{g}_w \\ \dot{\mathbf{s}}_{w,t} + TQ_{ws,t}(\mathbf{y}^a_{s,t+T} - \mathbf{v}^a_{s,t}) + T\mathbf{g}_w \\ \exp\left(-\frac{T}{2}(\mathbf{y}^\omega_{s,t+T} - \mathbf{b}^\omega_{s,t} - \mathbf{v}^\omega_{s,t})\right) \odot \mathbf{q}_{sw,t} \\ \mathbf{b}^\omega_{s,t} + \mathbf{v}^{b^\omega}_{s,t} \end{bmatrix} \quad (2)$$

with gravity vector g_w and quaternion exponential $\exp(v)^T = \begin{bmatrix}\cos\|v\| & (v^T/\|v\|)\sin\|v\|\end{bmatrix}$. The camera pose is obtained from the state vector with $\mathbf{c}_w = \mathbf{s}_w + Q_{ws}\mathbf{c}_s$ and $\mathbf{q}_{cw} = \mathbf{q}_{cs} \odot \mathbf{q}_{sw}$, where $\mathbf{q}_{cs}$ and $\mathbf{c}_s$ denote the hand-eye rotation and translation between the camera and the IMU, respectively (cf. Figure 2).

The MPF framework requires a partition of the state vector into a nonlinear and a linear part and a reformulation of the dynamic model according to (1a) and (1b). The gyroscope biases $\mathbf{b}_s^\omega$ and the velocities $\dot{\mathbf{s}}_w$ are eligible for the linear partition. However, as in the considered state-space model the Riccati recursions have to be evaluated for each particle, the MPF becomes less efficient, if many states are marginalised (cf. [27]). The gyroscope biases can be regarded as quasi-static parameters implying that the KF is the more appropriate estimation tool (cf. [32]). Moreover, adding these parameters to the nonlinear partition would result in a significantly increased demand for particles and random numbers per particle. Adding the velocities to the nonlinear partition, however, should not cause these effects due to their correlation with the position parameters. These considerations led to the marginalisation of the gyroscope biases only. Applying this partition and, in order to linearise the orientation update, the small angle approximation to equation (2) results in the following linear/nonlinear system dynamics:

$$\underbrace{\begin{bmatrix} \mathbf{s}_{w,t+T} \\ \dot{\mathbf{s}}_{w,t+T} \\ \mathbf{q}_{sw,t+T} \end{bmatrix}}_{\mathbf{x}^n_{t+T}} = \underbrace{\begin{bmatrix} \mathbf{s}_{w,t} + T\dot{\mathbf{s}}_{w,t} + \frac{T^2}{2}(Q_{ws,t}\mathbf{y}^a_{s,t+T} + \mathbf{g}_w) \\ \dot{\mathbf{s}}_{w,t} + T(Q_{ws,t}\mathbf{y}^a_{s,t} + \mathbf{g}_w) \\ \mathbf{q}_{sw,t} - \frac{T}{2}S(\mathbf{q}_{sw,t})\mathbf{y}^\omega_{s,t+T} \end{bmatrix}}_{f^n_t(\mathbf{x}^n_t, \mathbf{u}^n_{t+T})} + \underbrace{\begin{bmatrix} 0_{3\times3} \\ 0_{3\times3} \\ \frac{T}{2}S(\mathbf{q}_{sw,t}) \end{bmatrix}}_{A^n_t(\mathbf{x}^n_t)} \underbrace{\mathbf{b}^\omega_{s,t}}_{\mathbf{x}^l_t} + \underbrace{\begin{bmatrix} 0_{3\times3} & -\frac{T^2}{2}Q_{ws,t} \\ 0_{3\times3} & -TQ_{ws,t} \\ \frac{T}{2}S(\mathbf{q}_{sw,t}) & 0_{4\times3} \end{bmatrix}}_{G^n_t(\mathbf{x}^n_t)} \underbrace{\begin{bmatrix} \mathbf{v}^\omega_{s,t} \\ \mathbf{v}^a_{s,t} \end{bmatrix}}_{\mathbf{v}^n_t} \quad (3a)$$

$$\underbrace{\mathbf{b}^\omega_{s,t}}_{\mathbf{x}^l_{t+T}} = \underbrace{I_{3\times3}}_{A^l_t(\mathbf{x}^n_t)} \underbrace{\mathbf{b}^\omega_{s,t}}_{\mathbf{x}^l_t} + \underbrace{I_{3\times3}}_{G^l_t(\mathbf{x}^n_t)} \underbrace{\mathbf{v}^{b^\omega}_{s,t}}_{\mathbf{v}^l_t} \quad (3b)$$

with

$$S(q) = \begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & q_z & -q_y \\ -q_z & q_w & q_x \\ q_y & -q_x & q_w \end{bmatrix}.$$

The exact orientation update as given in (2) can still be used during step 4 of Algorithm 1 as nonlinearities are handled in the PF. This ensures that the sampled quaternions have unit length.

The observation model is now derived. For each timestamp t the vision sensor delivers a set of 2D/3D point correspondences $(\mathbf{m}_{n,t}, \mathbf{m}_{w,t})^{(j)}$ with measurement noises $\mathbf{e}^c_{n,t} \sim \mathcal{N}(\mathbf{0}_2, R^{(j)}_{nn,t})$ and $\mathbf{e}^c_{w,t} \sim \mathcal{N}(\mathbf{0}_3, R^{(j)}_{ww,t})$. The measurement equation for one such correspondence (j) is modelled implicitly as:

$$\underbrace{\mathbf{0}_2}_{\mathbf{y}^{(j)}_t} = \underbrace{\begin{bmatrix} I_2 & -\mathbf{m}_{n,t} \end{bmatrix} Q_{cs}\big(Q_{sw,t}(\mathbf{m}_{w,t} - \mathbf{s}_{w,t}) - \mathbf{c}_s\big)}_{h_t(\mathbf{x}^n_t, \mathbf{m}^{(j)}_{n,t}, \mathbf{m}^{(j)}_{w,t})} + \underbrace{\mathbf{e}^c_t}_{\mathbf{e}^{(j)}_t}, \quad (4a)$$

where Q_{cs} and $\mathbf{c}_s$ denote the hand-eye transformation as introduced in Section 3.2. In order to apply the MPF the pdf of the measurement noise $p(\mathbf{e}^{c,(j)}_t)$ is needed. As (4a) does not depend on the

[3] As in this case the state space includes no angular velocities, the orientation can only be predicted, when inertial readings are available.

[4] Given that the accelerometers measure not only free accelerations but also accelerations due to gravity, which have to be subtracted using the estimated orientation. For a detailed investigation of this issue the reader is referred to [16]. General information about the error characteristics of inertial sensors are given in [31].

linear states, this pdf can be arbitrary. However, if the proposed pose estimation method is used together with FastSLAM, the measurement noise needs to be Gaussian. Thus, a standard first order error propagation is used to approximate the covariance of the measurement noise $R_t^{(j)}$ given the covariances $R_{nn,t}^{(j)}$ and $R_{ww,t}^{(j)}$ and $\mathbf{h}_t = h_t(\mathbf{x}_t^n, \mathbf{m}_{n,t}^{(j)}, \mathbf{m}_{w,t}^{(j)})$:

$$\mathbf{e}_t^{c,(j)} \sim \mathcal{N}(\mathbf{0}_2, R_t^{(j)}) \text{ with} \tag{4b}$$

$$R_t^{(j)} \approx \begin{bmatrix} \frac{\partial \mathbf{h}_t}{\partial \mathbf{m}_n} & \frac{\partial \mathbf{h}_t}{\partial \mathbf{m}_w} \end{bmatrix} \begin{bmatrix} R_{nn,t}^{(j)} & 0_{2\times 3} \\ 0_{3\times 2} & R_{ww,t}^{(j)} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{h}_t}{\partial \mathbf{m}_n}^T \\ \frac{\partial \mathbf{h}_t}{\partial \mathbf{m}_w}^T \end{bmatrix}. \tag{4c}$$

Note that all equations in (3) and (4) refer to one single particle $[i]$. In order to compute the importance weights $\{w_t^{[i]}\}_{i=1}^N$ in step 1 of Algorithm 1, the likelihood $p(\mathbf{y}_t|\mathbf{x}_{0:t}^{n,[i]}, \mathbf{u}_{0:t}^n, \mathbf{y}_{0:t-T})$ has to be evaluated for each particle $[i]$. This is straightforward from (4) using the fact that the vision measurements can be assumed conditionally independent (cf. [32]). If resampling is not performed after each time step, the importance weights are updated multiplicatively:

$$w_t^{[i]} = w_{t-rT}^{[i]} \prod_{j=1}^{M} \mathcal{N}\big(\mathbf{0}_2; h_t(\mathbf{x}_t^{n,[i]}, \mathbf{m}_{n,t}^{(j)}, \mathbf{m}_{w,t}^{[i],(j)}), R_t^{[i],(j)}\big), \tag{5}$$

where M is the number of vision measurements, $t - rT$ is the time of the previous weight update and $R_t^{[i],(j)}$ corresponds to (4c). The 3D feature positions $\mathbf{m}_{w,t}^{[i],(j)}$ and covariances $R_t^{[i],(j)}$ are assumed to be different in each particle resulting in exactly the same likelihood computation as given for the FastSLAM algorithm (cf. [32]).

3.3 Extending the original MPF

All quantities needed for the MPF are now derived. However, Algorithm 1 is rather general. In order to match the considered application, a problem-specific reformulation, which accounts for the system architecture and workflow outlined in Figure 1 and exploits the multi-rate characteristics of the sensors, is given in Algorithm 2.

This algorithm produced reasonable results on simulated data. However, processing realistic inertial data and vision measurements resulted in an increased drift in the translational states and frequent filter divergences, at least with a particle set allowing real-time performance ($N \leq 100$). The observed drift is mainly caused by using the accelerometer readings as control inputs to the nonlinear time update on top of a coarse orientation prior. Errors in the prior orientation are again caused by a couple of reasons: the relatively high residual errors of the low-cost gyroscopes, errors in the hand-eye calibration and the usually somewhat excessive process noise needed by the PF to spread out the particles sufficiently during the time update. However, the accelerometers provide valuable information about the orientation and the translational parts of the state space that should be used. This implies that the orientation estimates in the particles and the overall robustness of the filter have to be improved. Therefore, Algorithm 2 has been extended with several concepts like automatic model-switching, improved orientation proposals, adaptive process noise and mixture proposals for the translational states. Some of these ideas have been used in different contexts, see for instance [6, 34–38].

3.3.1 Automatic model-switching and improved orientation proposals

The accelerometer measurement equation as adopted in (3a) is:

$$\mathbf{y}_{s,t}^a = Q_{sw,t}(\ddot{\mathbf{s}}_{w,t} - \mathbf{g}_w) + \mathbf{e}_{s,t}^a. \tag{6}$$

Algorithm 2: Visual-inertial MPF

Let $\langle \mathbf{x}_{t|t}^n, \hat{\mathbf{x}}_{t|t}^l, P_{t|t}, w_t \rangle^{[i]}$, $i = 1, \ldots, N$ be a set of properly weighted particles representing the posterior probability distribution at time t after the vision measurements have just been used. Assume that T is the sample time of the IMU and kT is the sample time of the vision sensor. The algorithm for processing the inertial control inputs up to and including the next set of vision measurements at timestep $t + kT$ is then given by the following steps:

1. For $r = 1, \ldots, k$:
   (a) PF TU: compute $\mathbf{x}_{t+rT|t+(r-1)T}^{n,[i]} \sim p(\mathbf{x}_{t+rT}^n|\mathbf{x}_{0:t+(r-1)T}^{n,[i]}, \mathbf{u}_{0:t+rT}^n, \mathbf{y}_{0:t+(r-1)T})$ based on (3a)
   (b) KF CMU: compute $p(\mathbf{x}_{t+(r-1)T}^{l,[i]}|\mathbf{x}_{0:t+rT}^{n,[i]}, \mathbf{u}_{0:t+(r-1)T}^l, \mathbf{y}_{0:t+(r-1)T})$ based on (3a)
   (c) KF TU: compute $p(\mathbf{x}_{t+rT}^{l,[i]}|\mathbf{x}_{0:t+rT}^{n,[i]}, \mathbf{u}_{0:t+rT}^l, \mathbf{y}_{0:t+(r-1)T})$ based on (3b)
2. Compute a pose prediction at time $t + kT$ for the image analysis.
3. If features were observed, compute the importance weights using (5) and normalise, otherwise omit the subsequent steps[a].
4. Resample using systematic resampling.[b]
5. In case of SLAM, update the 3D positions of the registered features $\mathbf{m}_{w,t}^{[i],(j)}$ in each particle using standard EKFs (cf. [1, 32]).

[a] Note, as the measurement equation (4a) does not depend on the linear states, step 3 of Algorithm 1 (KF MU) is omitted and all information enters the linear states $\mathbf{b}_s^\omega$ during the KF CMU in step 1b.

[b] This requires only one random number. Further resampling techniques are given for instance in [33].

If no (significant) body accelerations $\ddot{\mathbf{s}}_{w,t}$ are present it reduces to:

$$\mathbf{y}_{s,t}^a = -Q_{sw,t}\mathbf{g}_w + \mathbf{e}_{s,t}^a. \tag{7}$$

This equation can still be used to estimate the orientation, but obviously it contains no information about the translational states implying that no drift is introduced into these parameters. However, if significant body accelerations are present, model (6) provides better tracking quality. From these considerations it suggests itself to distinguish between a high and a low acceleration state-space model and to switch between both based on whether body accelerations are detected or not. A rather simple detection criterion has been adopted here (cf. [35]): the low acceleration model is used, if the magnitude of the accelerometer reading equals the magnitude of the gravity vector except for a threshold D^a for a certain amount of time D^T. The high acceleration model is given in Section 3.2. The goal of this paragraph is to develop a low acceleration model, which treats the accelerometer readings as measurements according to (7) and yields an improved orientation proposal. This is achieved by the following procedure performed separately for each particle $[i]$: at time t the quaternion sample $\mathbf{q}_{sw,t|t}^{[i]}$ is moved to the linear state

partition $\hat{\mathbf{x}}_{t|t}^{[i]}$ using:

$$\hat{\mathbf{x}}_{t|t}^{*l,[i]} = \begin{bmatrix} \mathbf{q}_{sw,t|t}^{[i]} \\ \hat{\mathbf{b}}_{s,t|t}^{\omega,[i]} \end{bmatrix} \text{ and } P_{t|t}^{*l,[i]} = \begin{bmatrix} 0_{4\times4} & 0_{4\times3} \\ 0_{3\times4} & P_{b^\omega b^\omega,t|t}^{[i]} \end{bmatrix}. \quad (8)$$

The low body acceleration assumption implies that a constant velocity model can be used for the nonlinear time update of the remaining translational states:

$$\begin{bmatrix} \mathbf{s}_{w,t+T} \\ \dot{\mathbf{s}}_{w,t+T} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{w,t} + T\dot{\mathbf{s}}_{w,t} \\ \dot{\mathbf{s}}_{w,t} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \mathbf{v}_{w,t}^{\ddot{s}}. \quad (9a)$$

The gyroscope reading is then used as control input $\mathbf{u}_{t+T}^{l} = \mathbf{y}_{s,t+T}^{\omega}$ to the time update of the linear states:

$$\begin{bmatrix} \mathbf{q}_{sw,t+T} \\ \mathbf{b}_{s,t+T}^{\omega} \end{bmatrix} = \begin{bmatrix} \exp\big(-\frac{T}{2}(\mathbf{y}_{s,t+T}^{\omega} - \mathbf{b}_{s,t}^{\omega} - \mathbf{v}_{s,t}^{\omega})\big) \odot \mathbf{q}_{sw,t} \\ \mathbf{b}_{s,t}^{\omega} + \mathbf{v}_{s,t}^{b^\omega} \end{bmatrix}. \quad (9b)$$

The nonlinear orientation update implies that an EKF is needed in each particle instead of a KF. The accelerometer reading $\mathbf{y}_{s,t+T}^{a}$ is used in the measurement update of the linear states (7). At time $t+T$ a quaternion is sampled from the posterior Gaussian:

$$\mathbf{q}_{sw,t+T|t+T}^{[i]} \sim \mathcal{N}(\hat{\mathbf{x}}_{t+T|t+T}^{*l,[i]}, P_{t+T|t+T}^{*l,[i]}) \quad (10a)$$

and moved back to the nonlinear partition. This affects the remaining linear states. Let

$$\begin{aligned} \hat{\mathbf{q}} &:= \hat{\mathbf{q}}_{sw,t+T|t+T}^{[i]} \\ \mathbf{q} &:= \mathbf{q}_{sw,t+T|t+T}^{[i]} \\ \hat{\mathbf{b}} &:= \hat{\mathbf{b}}_{s,t+T|t+T}^{\omega,[i]}. \end{aligned}$$

From the Gaussian conditioning operation it follows that:

$$\hat{\mathbf{b}}|\mathbf{q} \sim \mathcal{N}\big(\hat{\mathbf{b}} + P_{bq}P_{qq}^{-1}(\hat{\mathbf{q}} - \mathbf{q}), P_{bb} - P_{bq}P_{qq}^{-1}P_{qb}\big). \quad (10b)$$

The procedure above, which is similar to FastSLAM 2.0 [6], has some further implications compared to the high acceleration case: with the additional measurement (7), the EKF MU (step 3 of Algorithm 1) is required instead of the EKF CMU (step 5) as used in Algorithm 2. This affects the likelihoods of the particles, which hence have to be updated at IMU sampling rate T. In particular, the update formula is:

$$w_t^{[i]} = w_{t-T}^{[i]} \mathcal{N}(\mathbf{y}_{s,t}^{a}; \hat{Q}_{sw,t|t-T}^{[i]}\mathbf{g}_w, H_t^{[i]}P_{t|t-T}^{l,[i]}H_t^{[i]^T} + R_{ss}) \quad (11)$$

$$\text{with } H_t^{[i]} = \frac{\partial(\hat{Q}_{sw,t|t-T}^{[i]}\mathbf{g}_w)}{\partial(\mathbf{q}_{sw}, \mathbf{b}_s^{\omega})} \text{ and } \mathbf{e}_s^{a} \sim \mathcal{N}(\mathbf{0}_3, R_{ss}).$$

At time $t + kT$, when the next set of vision measurements arrives, (5) can be used as before.

3.3.2 Adaptive process noise

As the drift observed in the translational states when using the high acceleration model correlates with the body accelerations, it is self-evident to adapt the acceleration process noise $\mathbf{v}_{s,t}^{a}$ used in the PF TU (3a) depending on the magnitude of the predicted body accelerations $\hat{\ddot{s}}_{w,t|t-T}$. The adaption is done separately for each particle $[i]$ using:

$$\mathbf{v}_{s,t}^{a,[i]} = \mathbf{v}_{s,t}^{a} + \alpha \| \hat{\ddot{s}}_{w,t|t-T} \| \quad (12)$$

with $\hat{\ddot{s}}_{w,t|t-T} = \hat{Q}_{ws,t|t-T}\mathbf{y}_{s,t}^{a} + \mathbf{g}_w$. Values for α are given in Table 1. If the considered application does not require a constant sampling time as is preferred in our real-time case, the same mechanism can be used to adapt the number of particles.

3.3.3 Mixture proposal

The adaptive acceleration process noise improves the robustness of the filter and moderates the effects of the drift over a short period of time but does not solve the problem. The reason is that all particles might be affected depending on the quality of their respective orientation sample. In order to avoid this in the first place, a mixture proposal is used for the translational states: when the high acceleration model is applied, half of the particles are sampled according to the constant acceleration model given in (3a) and the rest are sampled using the constant velocity model given in (9a). This ensures the existence of both, particles that are not affected by a potential drift and others that are able to track high body accelerations, for instance due to shifts in the direction of movement. The adaptive process noise mechanism is also applied to the body acceleration process noise $\mathbf{v}_{w,t}^{\ddot{s}}$, which differs from the accelerometer process noise $\mathbf{v}_{s,t}^{a}$. The mixture proposal affects the nonlinear time update of the translational states, while all other equations given in this paper hold.

3.3.4 Final algorithm

The extensions presented in this section affect only the processing of the inertial data. The resulting Algorithm 3 therefore replaces steps 1a through 1c of Algorithm 2, while the rest of Algorithm 2 remains unaffected.

Algorithm 3

1. If high body accelerations are present: **High acceleration model**
   (a) PF TU:
      i. Sample the new orientation based on (3a).
      ii. Adapt the process noise for the translational states using (12).
      iii. Sample a mixture proposal for the translational states (Section 3.3.3).
   (b) KF CMU and KF TU based on (3a) and (3b).
2. Else: **Low acceleration model**
   (a) Shift the quaternion to the linear partition using (8).
   (b) PF and EKF TU based on (9).
   (c) PF and EKF MU update based on (11) and (7).
   (d) Shift the quaternion to the nonlinear partition using (10).

4 Experimental Setup and Results

The camera-IMU system used for the experiments combines a monochrome PGR camera with an XSens MT9-C IMU in one housing. Both devices are synchronised in hardware providing a synchronised stream of images ($25/12.5$ Hz, 320×240 resolution) and IMU data (100 Hz). The vision measurements are provided by a marker-based tracking system. One rectangular marker gives four 2D/3D point correspondences. Including the image analysis and visualisation, the system operates smoothly at 25 Hz camera framerate on a $2,2$ GHz laptop if $N \leq 100$ particles are used. Further details on the performance are given below. For an in-depth evaluation a challenging data sequence (synchronised images and IMU data) has been captured with some quick shifts in the direction of movement and erratic motions, which are characteristic for a handheld camera (cf. Figure 3). Based on this test sequence the

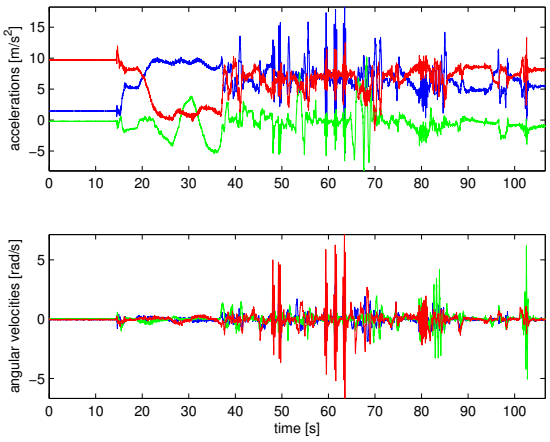


Figure 3: Acceleration and angular velocity signals of the test sequence in order to give an idea of the range of the performed movements.

precision, robustness and efficiency of the proposed pose estimation method, subsequently referred to as *MPF*, has been compared with previous visual-inertial fusion strategies introduced in Section 1. Lacking real ground truth data, the camera poses provided by the marker tracking system (*marker poses*) were used as reference for the results presented subsequently. These poses are obtained from the four 2D/3D point correspondences in a two-stage process based on the iterative POSIT algorithm with an extension to planar point configurations [39] followed by nonlinear least squares minimisation using the Levenberg-Marquard method [40]. The extraction of the marker corners is described in [41].

The marker poses were also used for initialising the filter and for reinitialising it after a divergence has been detected. The particle filter is said to diverge, if all particles have a negligible importance weight:

$$w_t^{[i]} < D^w \qquad \forall i \in [1 \dots N], \tag{13}$$

with $w_t^{[i]}$ from step 3 of Algorithm 2 before the normalisation. The system parameters and noise settings used during the experiments are summarised in Table 1.

4.1 Comparison with the EKF

For a comparison of the proposed method to the EKF the latter has been applied to the state-space model given in (2) and (4a). To give an impression of the computational complexity of both filters, the average computing time needed for processing the data of one frame — that is four inertial data at 25 Hz camera framerate and four vision measurements obtained from one marker — was measured with 0.51 ms for the EKF and 15.18 ms for the MPF. Considering these values and the fact that the current MPF implementation is not optimised, for instance by parallelisation, a FastSLAM system adopting the proposed method for pose estimation can be expected to run at a constant framerate of 12.5 Hz.

Figure 4 shows the trajectories obtained from the marker tracking system, the EKF and the MPF. The plots interfere with each other roughly, however, considering the marker poses as ground truth, the EKF provided a higher precision. The average absolute estimation errors are given in Table 2 (columns one and three). Note that the reference trajectory obtained from the marker tracking system is also subject to an unknown error. Qualitatively speaking, the EKF provided a smoother trajectory while the MPF tended to introduce jitter.

Table 1: System parameters and standard deviation noises — assuming equal noise in all dimensions — used during the experiments: as the marker corners are known by ground truth, the 3D feature locations were assumed certain. Note that the settings were tuned experimentally on base of the test sequences and that the quantitative results presented in this section of course depend on the chosen settings.

	MPF	EKF	PF	Complex MPF
Process noise				
$\mathbf{v}_{s,t}^{\omega}$	0.02	0.02	0.02	0.1
$\mathbf{v}_{s,t}^{a}$	0.5	0.5	–	–
$\mathbf{v}_{w,t}^{\ddot{s}}$	0.5	–	2.4	5.0
$\mathbf{v}_{s,t}^{b^\omega}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	–	$5 \cdot 10^{-4}$
$\mathbf{v}_{s,t}^{b^a}$	–	–	–	$1 \cdot 10^{-3}$
Meas. noise				
$\mathbf{e}_{s,t}^{\omega}$	–	–	–	0.02
$\mathbf{e}_{s,t}^{a}$	0.1	–	–	0.4
$\mathbf{e}_{n,t}$	$7 \cdot 10^{-3}$	$7 \cdot 10^{-3}$	$7 \cdot 10^{-3}$	$7 \cdot 10^{-3}$
System				
N	100	–	300	100
D^T (cf. 3.3.1)	0.4	–	–	–
D^a (cf. 3.3.1)	0.4	–	–	–
D^w (cf. (13))	$1 \cdot 10^{-15}$	–	$1 \cdot 10^{-15}$	$1 \cdot 10^{-15}$
α (cf. 3.3.2)	3	–	–	–

Table 2: Average absolute position and orientation (Euler angles) errors measured for the MPF, the EKF and the complex MPF.

	EKF		Complex MPF		MPF
Position [cm]					
Δx	0.26	$<$	0.46	$>$	0.38
Δy	0.26	$<$	0.47	$<$	0.80
Δz	0.27	$<$	0.52	$<$	0.69
Orientation [deg]					
Δx	0.57	$<$	0.72	$<$	1.43
Δy	0.45	$<$	0.59	$<$	0.91
Δz	0.33	$<$	0.51	$<$	0.95

Moreover, the ability of the proposed method to estimate the gyroscope biases has been investigated, this time using a data sequence with repeated movements and some stationary parts. The results are presented in Figure 5. The bias estimates obtained from the MPF method converge properly to approximately the same values as those obtained from the EKF even though the convergence is slower. Without the extensions described in Section 3.3.1 (automatic model-switching and improved orientation proposals) the convergence is significantly worse. The two major conclusions to draw from this figure are: first, the MPF method is able to estimate the gyroscope biases correctly, and second, the automatic change over to the low acceleration model has a positive effect on the estimation of the gyroscope bias parameters. This again improves the quality of the orientation samples and as a result the quality of the position samples, when the high acceleration model is used. The reason for the improved convergence is the accelerometer measurement update in step 2c of Algorithm 3. The accelerometer measure-

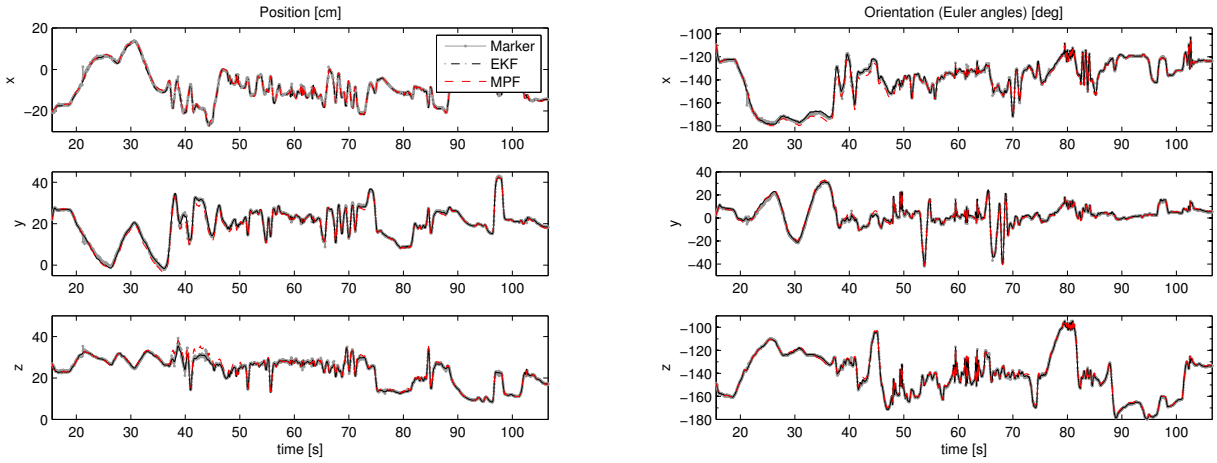


Figure 4: Position and orientation estimates obtained from the marker system, the EKF and the MPF.

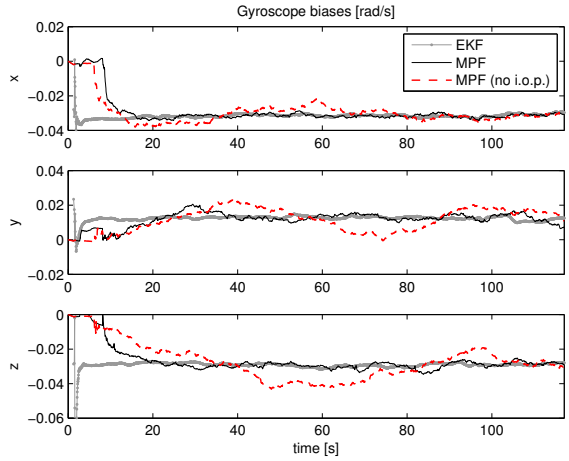


Figure 5: Gyroscope bias estimates obtained from the EKF, the MPF and the MPF without automatic model switching and improved orientation proposals (MPF no i.o.p.).

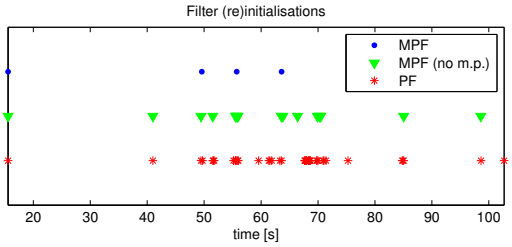


Figure 6: Filter reinitialisations required by the PF, the MPF and the MPF without the mixture proposal extension (MPF no m.p.): the latter means in this context, that all particles were sampled on base of the constant velocity model given in (9a), that is, the accelerometers were only used for adapting the process noise but not for sampling the translational states.

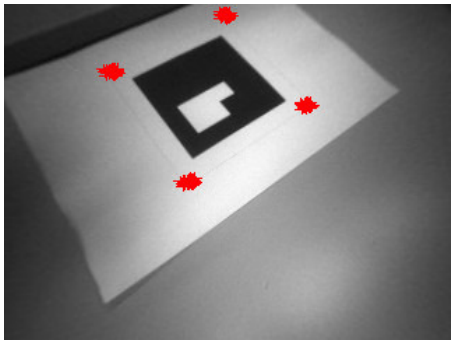
ments provide information about the orientation, which due to the correlations in the covariance matrix contributes to the estimation of the gyroscope biases. If the high acceleration model is always used, the biases are estimated only implicitly during the corrective KF measurement update (step 1b of Algorithm 3).

The results presented in this section clearly show the superiority of the EKF over the MPF, if real-time pose estimation is considered. However, the motivation for using the MPF is not given by the desire for improved pose estimates, but by the desire for a more scalable and robust solution in the context of SLAM. This idea is extended in Section 4.2.

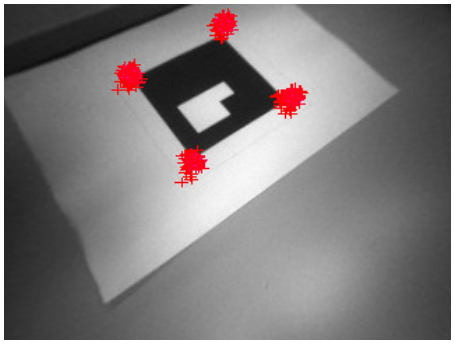
4.2 Comparison with the standard PF

In order to evaluate the benefits acquired by using the MPF for incorporating both types of inertial data, angular rates and accelerations, the proposed method has been compared with the simple PF based fusion model described in [17] (cf. Section 1). In [17] the angular rates are used for sampling the rotation angles, while a random walk is applied to the position. The time update for the orientation is effectively the same as in (2), whereas nothing is said about the sensor biases. Hence, those were fixed to reasonable values obtained from the EKF experiment above. In order to take the lower computational complexity of the PF method into account, $N = 300$ particles were allowed for this model. Thus, approximately the same amount of resources has been used by both methods. With the constant position model as proposed in [17], no reasonable results could be obtained on the considered test data. The constant velocity model given in (9a) performed better setting the body acceleration process noise $v^{\ddot{s}}_{w,t}$ to the average of the values obtained from the adaptive technique (cf. Section 3.3.2).

With the MPF using only one third of the particles used by the PF, the estimation errors obtained from both methods were comparable during successful filter operation. Concerning the orientation, this results from the fact, that both filters were initialised with the same gyroscope bias values and used the same orientation update rule, as the MPF operated mainly in the high acceleration mode on the considered test sequence. However, as presented in Figure 6, the PF showed continuous filter divergences, most of which arrived after shifts in the direction of movement and involved peaked position errors. In contrast, the MPF continued the tracking in most of the cases by exploiting the information given in the accelerometer measurements in the way developed in Section 3.3. In order to achieve a comparable robustness with the PF, a further increase of the process noise and a significant increase of the number of particles would be required. Aiming at real-time performance, the effects of increased particle numbers were not investigated in detail



(a) PF



(b) MPF

Figure 7: The overlaid crosses denote the features projected from the pose particles. (a) shows how the PF diverged after a change in the direction of motion, as no features project into the neighbourhood of the marker corners, implying that all particles have negligible weights. (b) demonstrates how the MPF continued the tracking, as the features project into the close neighbourhood of the marker corners. Note how the feature projections are spread out in (b) due to the automatic process noise increase described in Section 3.3.2.

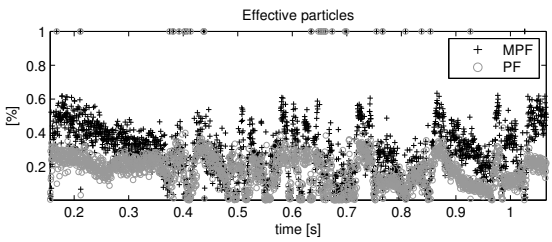


Figure 8: Percentage of effective particles from the MPF and the PF: note that the maximum values result from filter reinitialisations.

here. However, in the original paper, thousands of particles were used. The contribution of the mixture proposal extension to the robustness of the MPF shows also up in Figure 6 in terms of a reduced number of filter divergences. Figure 7 presents an exemplary frame from the test sequence, where the PF diverged, while the MPF was able to continue the tracking.

Figure 8 provides information about the efficiency of the PF and the MPF method in terms of the effective percentage of particles [5]:

$$P_{\text{eff},t} = \frac{1}{N \sum_{i=1}^{N} (w_t^{[i]})^2} \tag{14}$$

with $w_t^{[i]}$ from step 3 of Algorithm 2. $P_{\text{eff},t} \in [0 \dots 1]$ measures the percentage of particles contributing to the support of the estimated probability density function and determines the fraction of particles surviving the resampling process. It can be regarded as an indicator for the quality of the proposal distribution and, as computational power is spend when propagating particles in regions of low likelihood, for the efficiency of the filter. Figure 8 shows higher values for the MPF, implying that the survival times of the particles are extended and that the quality of the proposal distribution is improved compared to the PF. This is of special interest in the context of FastSLAM, not only because of computational reasons but also as the covariance information is lost with the particle trajectories. This implies that the average survival time of the particles is closely connected to the ability to close loops.

To summarise the results presented in this section, the MPF shows improved results with respect to robustness and filter efficiency by exploiting the accelerometer measurements in the way developed in Section 3.3. Moreover, it requires less particles than the PF for achieving the same precision, though the computational demand per particle is increased.

The ability to detect high body accelerations in advance is of further interest in the SLAM context. For instance, it can be used to prevent possible corruptions of the map states by prohibiting feature initialisations or map updates in case of high accelerations, where the pose samples are not likely to be well distributed. While leaping the map update is easy in FastSLAM, it cannot be done easily in the EKF, where the camera pose and the features are estimated jointly in one state. This is besides the better scalability a big advantage of FastSLAM over the EKF.

4.3 Comparison with the general MPF

Finally, the real-time capable pose estimation method described in this paper has been compared to the pose estimation part of the SLAM system proposed in [19] (cf. Section 1). In [19] the general MPF (Algorithm 1) is applied to the complex state-space model mentioned in Section 3.2 and the accelerometer biases are also estimated online. This method is referred to as *complex MPF*, whereas the proposed strategy is termed *MPF*. The noise settings used for the complex MPF are included in Table 1. Note that the tuning is more complex for this model, as treating the inertial data as measurements requires two additional noise processes.

To give an impression of the computational demands of the complex MPF, the average computing time needed for processing the data of one frame has been measured with 83.64 ms. 15.18 ms were measured for the proposed method yielding a reduction of the computational costs by a factor of 5.5.

Figure 9 shows the absolute estimation errors obtained from the complex MPF and the MPF on the considered test sequence. The average quantities are included in Table 2. Figure 10 presents the number of filter reinitialisations required by both methods. The complex MPF performed better with respect to the estimation precision. However, the MPF yielded more stable tracking results by requiring only four filter reinitialisations, while the complex MPF showed continuous divergences appearing mainly during combined rotational and translational camera motions. This experience is similar to what has been described for Algorithm 2 in the beginning of Section 3.3. Note that the results with respect to the estimation precision are qualified by the fact that the estimation errors are reset with each filter reinitialisation.

Figure 11 presents the percentage of effective particles for both methods. The complex MPF shows higher values, as the motion model is more predictive compared to the MPF, which uses the mixture proposal for the translational states and the adaptive process noise. However, a trade-off is given between the efficiency and the robustness of the filter.

To summarise the results of this section, the MPF provides reduced computational demands and an improved tracking stability,

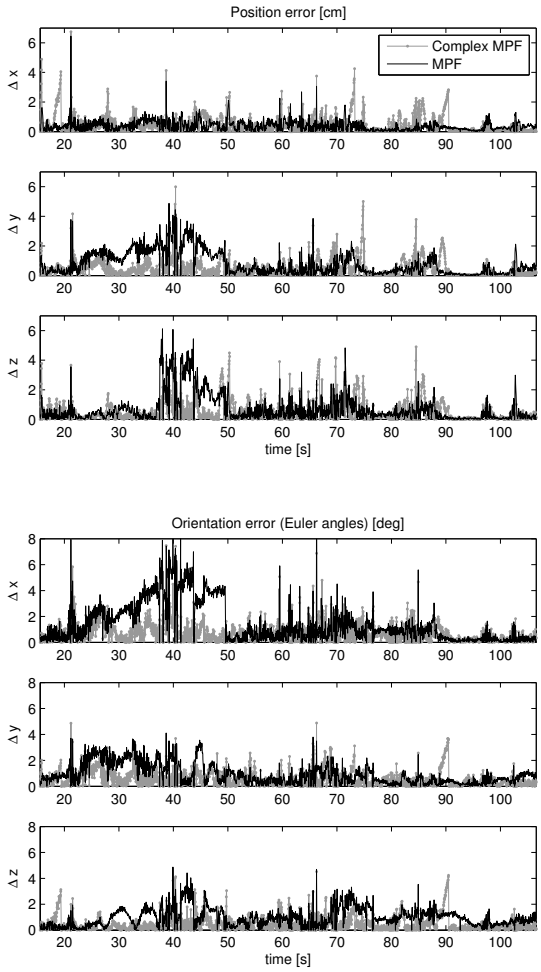


Figure 9: Absolute position and orientation errors obtained from the complex MPF and the MPF.

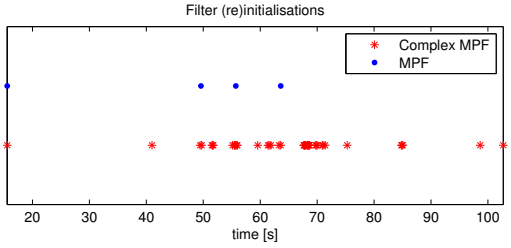


Figure 10: Filter reinitialisations required by the complex MPF and the MPF.

while the complex MPF yields a higher precision and filter efficiency. In the original paper [19], the camera-IMU system was mounted onto a robotic arm, which performed accurate continuous motions, while the test sequence considered here contains erratic motions, which are, however, specific for a handheld camera.

5 Conclusion and Future Work

This paper presents a strategy capable of fusing vision-based and inertial tracking technologies in real-time using the marginalised particle filter (MPF). The proposed method has been developed

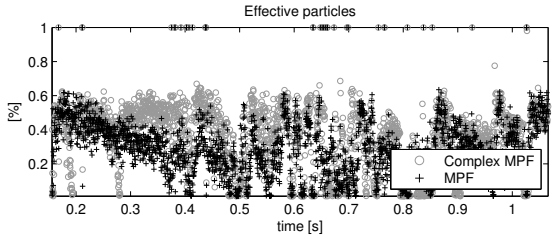


Figure 11: Percentage of effective particles from the complex MPF and the MPF.

in several steps. First, a mixed linear/nonlinear state-space model has been designed for the problem of real-time visual-inertial camera pose estimation and the MPF algorithm has been adapted to the specific characteristics of the considered application. Then the MPF has been extended with several concepts like automatic model-switching, improved orientation proposals, adaptive process noise and mixture proposals for the translational states. The performance of the final method has been evaluated in combination with a marker-based tracking system. It has been shown that the developed extensions increase the precision, the efficiency and most notably the robustness of the filter and that more stable results are obtained compared to previous fusion models based upon the standard particle filter and the general marginalised particle filter, especially in the presence of erratic camera motions. The extended Kalman filter was found to be superior to the fusion strategy based on the MPF with respect to precision, robustness and computational complexity. However, this holds for the pose estimation in a known environment, whereas the aim of this paper was to provide a real-time capable visual-inertial pose estimation algorithm, which can be used together with FastSLAM. The method presented here is completely compatible to FastSLAM and can be regarded as an extension of that, if inertial sensors are available.

Future work will consist of integrating the proposed visual-inertial pose estimation method with the markerless feature tracking and online reconstruction methods presented in earlier works [22, 23].

References

[1] M. Montemerlo and S. Thrun. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002.

[2] M. Montemerlo and S. Thrun. Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, Taiwan, 2003.

[3] R. Sim, P. Elinas, M. Griffin, A. Shyr, and J. J. Little. Design and analysis of a framework for real-time vision-based SLAM using Rao-Blackwellised particle filters. In *Canadian Conference on Computer and Robotic Vision (CRV)*, Quebec City, QC, 2006.

[4] E. Eade and T. Drummond. Scalable Monocular SLAM. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 469–476, New York, NY, June 2006.

[5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. 50(2):174–188, February 2002.

[6] M. Montemerlo and S. Thrun. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003.

[7] M. Pupilli and A. Calway. Real-time Camera Tracking Using a Particle Filter. In *British Machine Vision Conference (BMVC)*, pages 519–528, Oxford, England, September 2005. Department of Computer Science, University of Bristol, BMVA Press.

[8] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Atlantic City, NJ, February 2000.

[9] M. Pupilli. *Particle Filtering for Real-time Camera Localisation*. PhD thesis, University of Bristol, October 2006.

[10] G. Qian, R. Chellappa, and Q. Zheng. Robust Structure from Motion Estimation Using Inertial Data. *Optical Society of America*, 18:2982–2997, December 2001.

[11] E. Foxlin and L. Naimark. VIS-Tracker: A Wearable Vision-Inertial Self-Tracker. In *IEEE Virtual Reality (VR)*, Los Angeles, CA, March 2003.

[12] B. Jiang, U. Neumann, and S. You. A Robust Hybrid Tracking System for Outdoor Augmented Reality. In *IEEE Virtual Reality Conference (VR)*, Chicago, Illinois, USA, March 2004.

[13] G. Reitmayr and T. Drummond. Going out: Robust Model-based Tracking for Outdoor Augmented Reality. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 109–118, Santa Barabara, CA, October 2006.

[14] P. Pinies, T. Lupton, S. Sukkarieh, and J. D. Tardos. Inertial Aiding of Inverse Depth SLAM using a Monocular Camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, Rome, Italy, April 2007.

[15] J. Hol, T. Schön, H. Luinge, P. Slycke, and F. Gustafsson. Robust Real-Time Tracking by Fusing Measurements from Inertial and Vision Sensors. *Journal of Real-Time Image Processing*, 2(2-3):149–160, November 2007.

[16] G. Bleser and D. Stricker. Advanced tracking through efficient image processing and visual-inertial sensor fusion. In *IEEE Virtual Reality Conference (VR)*, Reno, Nevada, March 2008.

[17] G. Qian, R. Chellappa, and Q. Zheng. Bayesian structure from motion using inertial information. In *International Conference on Image Processing (ICIP)*, Rochester, NY, USA, September 2002.

[18] A. Fakhr-eddine and M. Malik. Inertial and vision head tracker sensor fusion using a particle filter for augmented reality systems. In *International Symposium on Circuits and Systems (ISCAS)*, Vancouver BC, Canada, May 2004.

[19] T. Schön, R. Karlsson, D. Törnqvist, and F. Gustafsson. A Framework for Simultaneous Localization and Mapping Utilizing Model Structure. In *International Conference on Information Fusion*, Quebec, Canada, July 2007.

[20] R. Karlsson, T. B. Schön, D. Törnqvist, G. Conte, and F. Gustafsson. Utilizing Model Structure for Efficient Simultaneous Localization and Mapping for a UAV Application. In *IEEE Aerospace Conference*, Big Sky, MT, USA, March 2008.

[21] T. Schön, F. Gustafsson, and P.-J. Nordlund. Marginalized Particle Filters for Mixed Linear Nonlinear State-Space Models. 3(7):2279–2289, July 2005.

[22] G. Bleser, H. Wuest, and D. Stricker. Online Camera Pose Estimation in Partially Known and Dynamic Scenes. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 56–65, Santa Barabara, CA, October 2006.

[23] G. Bleser, M. Becker, and D. Stricker. Real-time vision-based tracking and reconstruction. *Journal of Real-Time Image Processing*, 2(2-3):161–175, November 2007.

[24] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Radar and Signal Processing*, pages 107–112, Bombay, India, January 1993.

[25] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[26] A. Doucet, N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. pages 613 – 624, 2001.

[27] R. Karlsson, T. Schön, and F. Gustafsson. Complexity Analysis of the Marginalized Particle Filter. 53(11):4408–4411, November 2005.

[28] T. Schön, R. Karlsson, and F. Gustafsson. The Marginalized Particle Filter in Practice. In *IEEE Aerospace Conference*, Big Sky, USA, March 2006.

[29] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis*. Springer, 1992.

[30] J. Kim and S. Sukkarieh. Real-time implementation of airborne inertial-SLAM. *Robotics and Autonomous Systems*, 55:62–71, 2007.

[31] D. Titterton and J. Weston. *Strapdown Inertial Navigation Technology*. American Institute of Aeronautics and Astronautics, 2 edition, 2004.

[32] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[33] J. Hol, T. Schön, and F. Gustaffson. On Resampling Algorithms for Particle Filters. In *Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK, September 2006.

[34] M. Isard and A. Blake. A mixed-state Condensation tracker with automatic model-switching. In *IEEE International Conference on Computer Vision (ICCV)*, Bombay, India, January 1998.

[35] H. Rehbinder and X. Hu. Drift-free attitude estimation for accelerated rigid bodies. In *IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, May 2001.

[36] T. Harada, T. Mori, and T. Sato. Development of a Tiny Orientation Estimation Device to Operate under Motion and Magnetic Disturbance. *The International Journal of Robotics Research*, 26(6):547–559, 2007.

[37] G. Qian and R. Chellappa. Structure From Motion Using Sequential Monte Carlo Methods. *Computer Vision*, 2:614–621, 2001.

[38] T. Schön, A. Eidehall, and F. Gustafsson. Lane Departure Detection for Improved Road Geometry Estimation. In *IEEE Intelligent Vehicles Symposium*, pages 546–551, Tokyo, June 2006.

[39] D. Oberkampf, D. DeMenthon, and L. Davis. Iterative Pose Estimation using Coplanar Feature Points. *Computer Vision and Image Understanding*, 63(3):495 – 511, May 1996.

[40] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3 edition, 2007.

[41] H. Kato and M. Billinghurst. Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System. In *International Workshop on Augmented Reality*, page 8594, San Francisco, USA, October 1999.