Visual-Inertial 2D Feature Tracking based on an Affine Photometric Model

Dominik Aufderheide, Gerard Edwards and Werner Krybus

Abstract The robust tracking of point features throughout an image sequence is one fundamental stage in many different computer vision algorithms (e.g. visual modelling, object tracking, etc.). In most cases, this tracking is realised by means of a feature detection step and then a subsequent re-identification of the same feature point, based on some variant of a template matching algorithm. Without any auxiliary knowledge about the movement of the camera, actual tracking techniques are only robust for relatively moderate frame-to-frame feature displacements. This paper presents a framework for a visual-inertial feature tracking scheme, where images and measurements of an inertial measurement unit (IMU) are fused in order to allow a wider range of camera movements. The inertial measurements are used to estimate the visual appearance of a feature's local neighbourhood based on a affine photometric warping model.

1 Introduction

Many different applications in the field of computer vision (CV) require the robust identification and tracking of distinctive feature points in monocular image sequences acquired by a moving camera. Prominent examples of such applications are 3D scene modelling following the structure-from-motion (SfM) principle or the simultaneous localisation and mapping (SLAM) for mobile robot applications. The general procedure of feature point tracking can be subdivided in two distinctive phases:

D. Aufderheide (✉) · W. Krybus
Division Soest, Institute for Computer Science, Vision and Computational Intelligence,
South Westphalia University of Applied Sciences, Luebecker Ring 2, 59494 Soest, Germany
e-mail: aufderheide@fh-swf.de

W. Krybus
e-mail: krybus@fh-swf.de

G. Edwards
Department of Electronic & Electrical Engineering,
Faculty of Science and Engineering, The University of Chester,
Thornton Science Park, Pool Lane, Ince, Chester CH2 4NU, UK
e-mail: gerard.edwards@chester.ac.uk

© Springer International Publishing Switzerland 2015

J. M. R. S. Tavares, R. Natal Jorge (eds.), *Developments in Medical Image Processing and Computational Vision,* Lecture Notes in Computational Vision and Biomechanics 19,
DOI 10.1007/978-3-319-13407-9_18

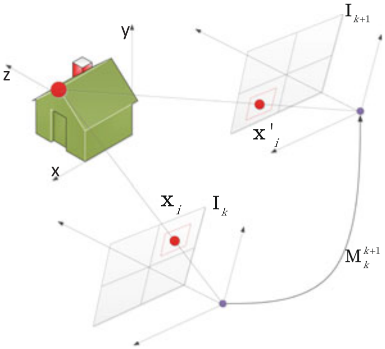


Fig. 1 Re-identification of single feature point in two subsequent frames of an image sequence

- *Detection*—The first stage is the identification of a set of distinctive point features ${}^kX = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ with $\mathbf{x}_i = (x, y)^T$ in image $\mathbf{I}_k$, e.g. based on computing the *cornerness* of each pixel (see [5]). At this stage each feature point is typically assigned with some kind of a descriptor $\theta\left(\mathbf{I}_{k(\mathbf{x}_i)}\right)$, which is used in the second stage for the re-identification of the feature. This descriptor could be a simple local neighbourhood of pixels around $\mathbf{x}_i$ or a more abstract descriptor such as the SIFT/SURF descriptors described by [9].
- *Re-identification*—The general task of feature tracking is the successful re-identification of the initial set of features ${}^kX$ from image $\mathbf{I}_k$ in the subsequent frame $\mathbf{I}_{k+1}$. Generally this can be described as an optimisation problem where the distance between a descriptor for pixel $\mathbf{x}'$ from $\mathbf{I}_{k+1}$ and the given descriptor $\theta\left(\mathbf{I}_{k(\mathbf{x}_i)}\right)$ should be minimised by varying $\mathbf{x}'$ within the image boundaries. In most cases the optimisation problem is not just driven by varying the image coordinates, but also by using some kind of a motion model $\Omega\left[\theta\left(\mathbf{I}_{k(\mathbf{x}_i)}\right)\right]_{\mathbf{M}_k^{k+1}}$ which tries to compensate the change in the descriptors appearance based on an estimation of the cameras movement $\mathbf{M}_k^{k+1}$ between $\mathbf{I}_k$ and $\mathbf{I}_{k+1}$. In order to reduce the computational complexity of the minimisation the range for varying both the pixel coordinates and the motion model parameters are limited to certain *search regions*. The general procedure of feature tracking is visualised in Fig. 1.

As it was shown by Aufderheide et al. [2], there are many ways for a feature tracking method to fail completely or produce a non-negligible number of incorrect matches. This can be clearly seen from a mathematical point of view by the fact that either the optimisation problem converges within a local minimum or not at all.

In Aufderheide et al. [1], we described a general approach for the combination of visual and inertial measurements within a parallel multi-sensory data fusion network for 3D scene reconstruction called *VISrec!*. Closely related to this work is the adaptation of ideas presented by Hwangbo et al. [6] for using the inertial measurements not only as an aiding modality during the estimation of the cameras egomotion, but also during the feature tracking itself.

The first stage for realising this was the development of an inertial smart sensor system (S^3) based on a bank of inertial measurement units in MEMS[1] technology. The S^3 is able to compute the actual absolute camera pose (position and orientation) for each frame. The hardware employed and the corresponding navigation algorithm are described in Sect. 2. As a second step a visual feature tracking algorithm, as described in Sect. 3, needs to be implemented. This algorithm considers prior motion estimates from the inertial S^3 in order to guarantee a greater convergence region of the optimisation problem and deliver an improved overall tracking performance. The results are briefly discussed in Sect. 4. Finally Sect. 5 concludes the whole work and describes potential future work.

2 Inertial Smart Sensor System S^3

For the implementation of an Inertial Fusion Cell (IFC) a smart sensor system (S^3) is suggested here, which is composed as a bank of different micro-electromechanical systems (MEMS). The proposed system contains accelerometers, gyroscopes and magnetometers. All of them are sensory units with three degrees of freedom (DoF). The S^3 contains the sensors itself, signal conditioning (filtering) and a multi-sensor data fusion (MSDF) scheme for pose (position and orientation) estimation.

2.1 General S^3 Architecture

The general architecture of the S^3 is shown in the following Fig. 2, where the overall architecture contains the main ‘organ’ consisting of the sensory units as described in Sect. 2.2. A single micro controller is used for analogue-digital-conversion (ADC), signal conditioning (SC) and the transfer of sensor data to a PC. The actual sensor fusion scheme is realised on the PC.

2.2 Hardware

The hardware setup of the S^3 is inspired by the standard configuration of a multi-sensor orientation system (MODS) as defined in [13]. The used system consists of a LY530AL single-axis gyro and a LPR530AL dual-axis gyro both from STMicroelectronics, which are measure the rotational velocities around the three main axis of the inertial coordinate system ICS (see Fig. 3). The accelerations of translational movements are measured by a triple-axis accelerometer ADXL345 from Analog Devices. Finally a 3-DoF magnetometer from Honeywell (HMC5843) is used to measure

[1] MEMS—micro-electromechanical systems.

Fig. 2 General architecture of the inertial S^3

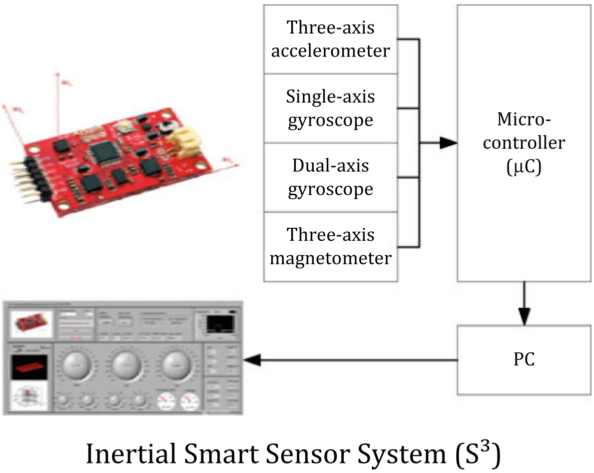
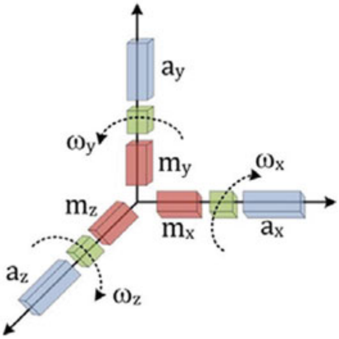


Fig. 3 General architecture of the inertial measurement units and measured entities



the earth's magnetic field. All IMU sensors are connected to a micro controller (ATMega328) which is responsible for initialisation, signal conditioning and communication. The interface between sensor and micro controller is based on I^2C-Bus for the accelerometer and magnetometer, while the gyroscope is directly connected to ADC channels of the AVR. So the used sensor setup consists of three orthogonal arranged accelerometers measuring a three dimensional acceleration $\mathbf{a}^b = \left[a_x\, a_y\, a_z\right]^T$ normalised with the gravitational acceleration constant g. Here b indicates the actual body coordinate system in which the entities are measured. The triple-axis gyroscope measures the corresponding angular velocities $\boldsymbol{\omega}^b = \left[\omega_x\, \omega_y\, \omega_z\right]^T$ around the sensitivity axes of the accelerometers. The magnetometer is used to sense the earth's magnetic field $\mathbf{m}^b = \left[m_x\, m_y\, m_z\right]^T$. Figure 3 shows the general configuration of all sensory units and the corresponding measured entities.

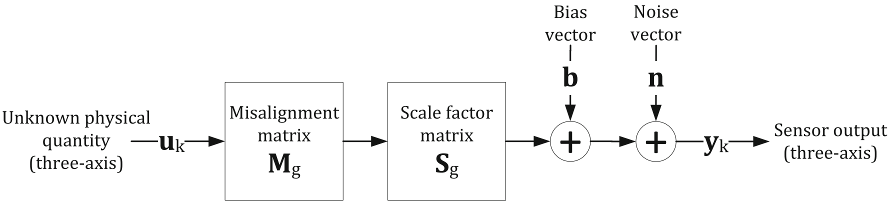


Fig. 4 General sensor model

2.3 Sensor Modelling and Signal Conditioning

Measurements from MEMS devices in general and inertial MEMS sensors in particular suffer from different error sources. Due to this it is necessary to implement both: an adequate calibration framework and a signal conditioning routine. The calibration of the sensory units is only possible if a reasonable sensor model is available in advance. The sensor model should address all possible error sources. Here the proposed model from [14] was utilised and adapted for the given context. It contains:

- Misalignment of sensitivity axes—Ideally the three independent sensitivity axes of each inertial sensor should be orthogonal. Due to imprecise construction of MEMS-based IMUs this is not the case for the vast majority of sensory packages. The misalignment can be compensated by finding a matrix **M** which transforms the non-orthogonal axis to a orthogonal setup.
- Biases—The output of the gyroscopes and accelerometers should be exactly zero if the S^3 is not moved at all. However there is typically a time-varying offset for real sensors. It is possible to differentiate *g-independent* biases (e.g. for gyroscopes) and *g-dependent* biases. For the latter there is a relation between the applied acceleration and the bias. The bias is modelled by incorporation of a bias vector **b**.
- Measurement noise—The general measurement noise has to be taken into account. The standard sensor model contains a white noise term **n**.
- Scaling factors—In most cases there is an unknown scaling factor between the measured physical quantity and the real signal. The scaling can be compensated for by introducing a scale matrix $\mathbf{S} = diag\left(s_x, s_y, s_z\right)$.

A block-diagram of the general sensor model is shown in the following figure (Fig. 4).

Based on this it is possible to define three separate sensor models for all three sensor types[2], as shown in the following equations:

$$\boldsymbol{\omega}_b = \mathbf{M}_g \cdot \mathbf{S}_g \cdot \boldsymbol{\omega}'_b + \mathbf{b}_g + \mathbf{n}_g \tag{1}$$

$$\mathbf{a}_b = \mathbf{M}_a \cdot \mathbf{S}_a \cdot \mathbf{a}'_b + \mathbf{b}_a + \mathbf{n}_a \tag{2}$$

[2] The different sensor types are indicated by the subscript indices at the entities in the different equations.

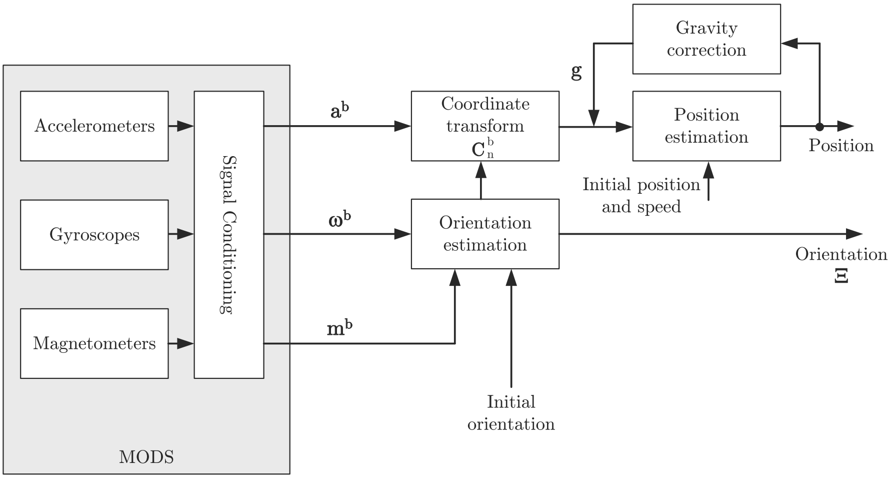

Gravity correction
g
Accelerometers
Gyroscopes
Magnetometers
Signal Conditioning
$\mathbf{a}^{\mathrm{b}}$
$\boldsymbol{\omega}^{\mathrm{b}}$
$\mathbf{m}^{\mathrm{b}}$
Coordinate transform $\mathrm{C}_{\mathrm{n}}^{\mathrm{b}}$
Position estimation
Position
Initial position and speed
Orientation estimation
Orientation Ξ
Initial orientation
MODS

Fig. 5 Computational elements of an INS

$$\mathbf{m}_b = \mathbf{M}_m \cdot \mathbf{S}_m \cdot \mathbf{m}'_b + \mathbf{b}_m + \mathbf{n}_m \tag{3}$$

It was shown that **M** and **S** can be determined by sensor calibration procedure in which the sensor array is moved to different known locations to determine the calibration parameters. Due to their time-varying character, the noise and bias terms cannot be determined a-priori. The signal conditioning step on the μC takes care of the measurement noise by integrating an FIR digital filter structure. The implementation realises a low-pass FIR filter based on the assumption that the frequencies of the measurement noise are much higher than the frequencies of the signal itself. The complete filter was realised in software on the μC, where the cut-off-frequencies for the different sensory units were determined by an experimental evaluation.

2.4 Basic Principles of Inertial Navigation

Classical approaches for inertial navigation are stable-platform systems which are isolated from any external rotational motion by specialised mechanical platforms. In comparison to those classical stable platform systems, the MEMS sensors are mounted rigidly to the device (here: the camera). In such a strapdown system, it is necessary to transform the measured quantities of the accelerometers, into a global coordinate system by using known orientations computed from gyroscope measurements. In general the mechanis system level operation of a strapdown inertial navigation systems (INS) can be described by the computational elements indicated in Fig. 5. The main problem with this classical framework is that location is determined by integrating measurements from gyros (orientation) and accelerometers (position). Due to superimposed sensor drift and noise, which is especially significant for MEMS devices, the errors for the egomotion estimation tend to grow unbounded.

The necessary computation of the orientation ξ of the S^3 based on the gyroscope measurements ω_b and a start orientation $\xi_{(t_0)}$ can be described as follows:

$$\xi = \xi_{(t_0)} + \int \omega_b dt \tag{4}$$

The integration of the measured rotational velocities would lead to an unbounded drifting error in the absolute orientation estimates. Figure 6 shows two examples for this typical drifting behaviour for all three Euler angles. For the two experiments shown in Fig. 6, the S^3 was not moved, but even after a short period of time (here: $6000 \cdot 0.01s = 60s$) there is an absolute orientation error of up to 4° clearly recognisable. For the estimation of the absolute position these problems are even more severe, because the position ϕ can be computed from acceleration measurements, in the inertial reference frame $\mathbf{a}_i$, only by double integration:

$$\phi = \phi_{(t_0)} + \int\int \mathbf{a}_i dt \tag{5}$$

Possible errors in the orientation estimation stage would lead also to a wrong position, due to the necessity to transform the accelerations in the body coordinate frame $\mathbf{a}_b$ to the inertial reference frame (here indicated by the subscript i).

The following figure (Fig. 7) demonstrates the typical drifting error for the absolute position (one axis) computed by using the classical strapdown methodology.

By using only gyroscopes, there is actually no way to control the drifting error for the orientation in a reasonable way. It is necessary to use other information channels. So the final framework for pose estimation considers two steps: an orientation estimation and a position estimation as shown in Fig. 8. In comparison to the classical strapdown method, the suggested approach here incorporates also the accelerometers for orientation estimation. The suggested fusion network is given in the following figure, and the different sub-fusion processes are described in Sects. 2.5 and 2.6.

2.5 Fusion for orientation

The general idea for compensating the drift error of the gyroscopes is based on using the accelerometers as an additional attitude sensor. Due to the fact that the 3-DoF accelerometer measures not only (external) translational motion, but also the influence of the gravity, it is possible to calculate the attitude based on the single components of the measured acceleration. At this point it should be noted that measurements from the accelerometers can only provide roll and pitch angle Thus, the heading angle of the unit has to be derived by using the magnetometer instead.

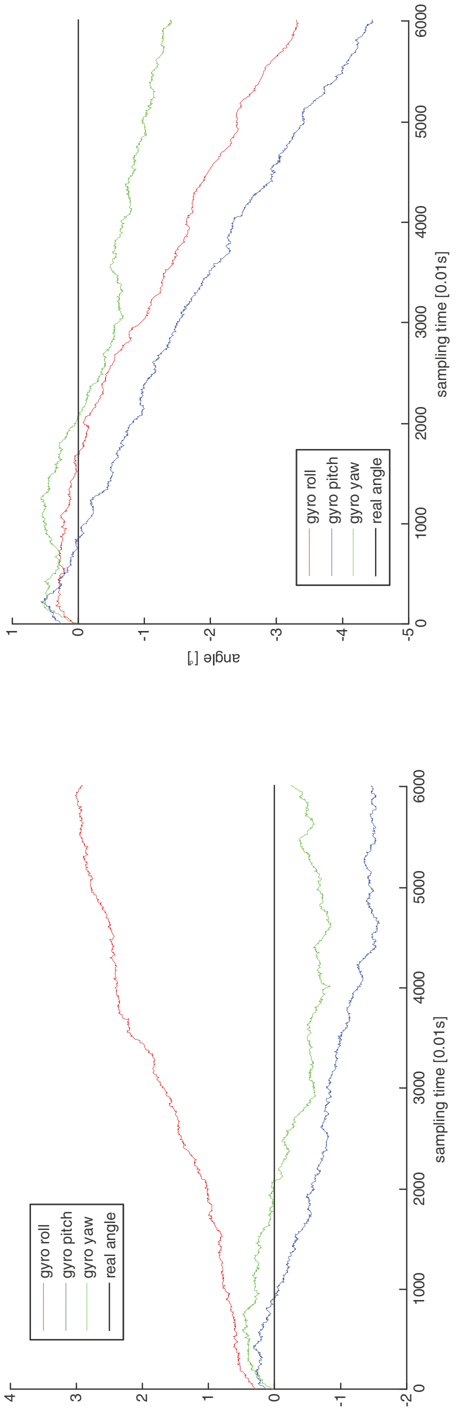


Fig. 6 Drifting error for orientation estimates based on gyroscope measurements, for two separate experiments

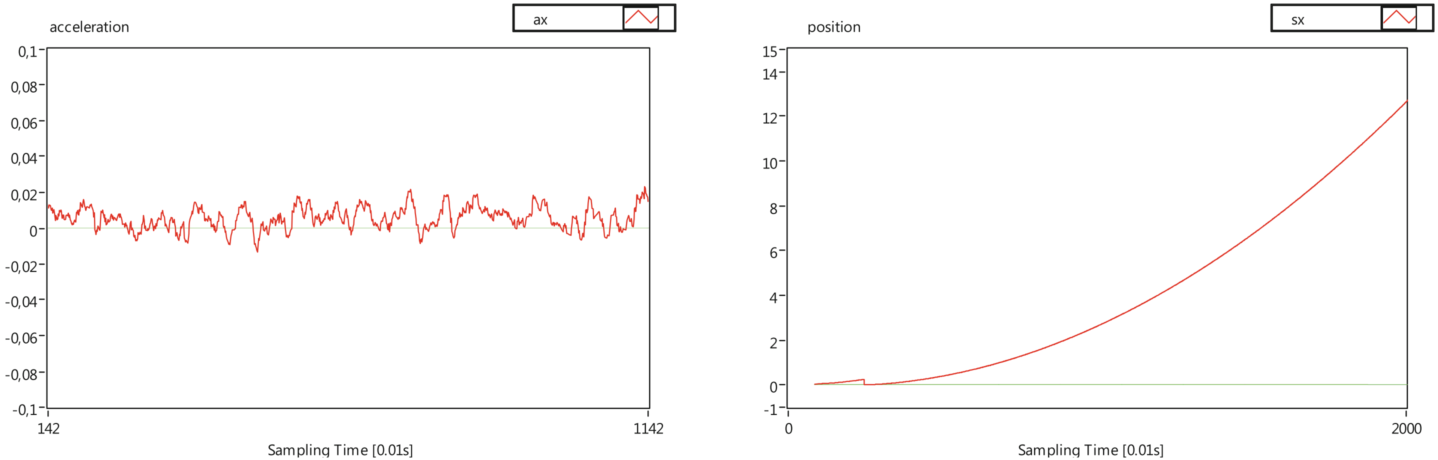


Fig. 7 Drifting error for absolute position estimates based on classical strapdown mechanisation of an inertial navigation system (*left*: acceleration measurements; *right*: absolute position estimate)

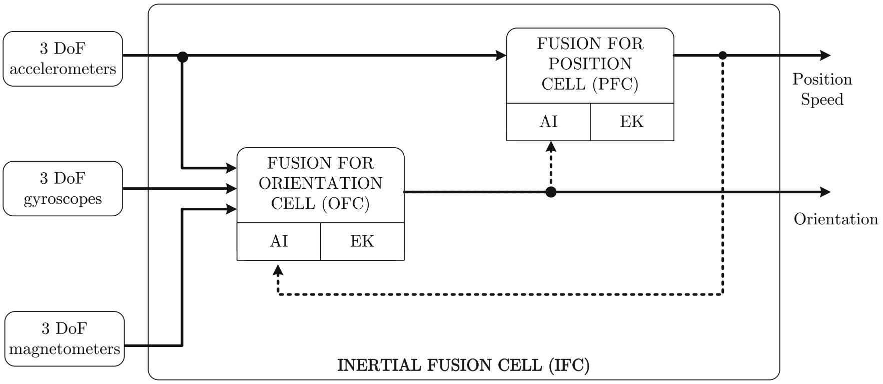

3 DoF accelerometers
3 DoF gyroscopes
3 DoF magnetometers
FUSION FOR POSITION CELL (PFC)
AI
EK
FUSION FOR ORIENTATION CELL (OFC)
AI
EK
Position
Speed
Orientation
INERTIAL FUSION CELL (IFC)

Fig. 8 System design of the inertial fusion cell (IFC)

Fig. 9 Geometrical relations between measured accelerations due to gravity and the roll and pitch angle of the attitude

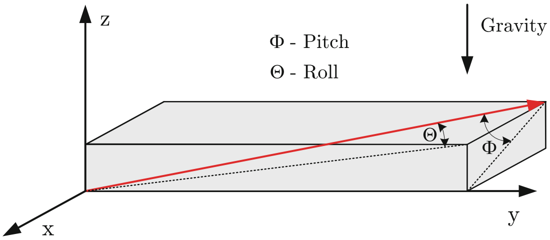


Figure 9 gives an illustration showing the geometrical relations between measured accelerations due to gravity and the roll and pitch angle of the attitude. The angles can be determined by following relations:

$$\theta = arctan2\left(a_x^2, \sqrt{(a_y + a_z)^2}\right) \tag{6}$$

$$\phi = arctan2\left(a_y^2, \sqrt{(a_x + a_z)^2}\right) \tag{7}$$

The missing heading angle can be obtained by using the readings from the magnetometer and the already determined roll and pitch angles. Here it is important to be aware that the measured elements of the earth magnetic field have to be transformed to the local horizontal plane (tilt compensation is illustrated in Fig. 10) as indicating in the corresponding relations

$$\begin{aligned} X_h &= m_x \cdot c\varphi + m_y \cdot s\theta \cdot s\varphi - m_z \cdot s\theta \cdot s\varphi \\ Y_h &= m_y \cdot c\theta + m_z \cdot s\theta \\ \psi &= \text{arctan}\,2\,(Y_h, X_h) \end{aligned} \tag{8}$$

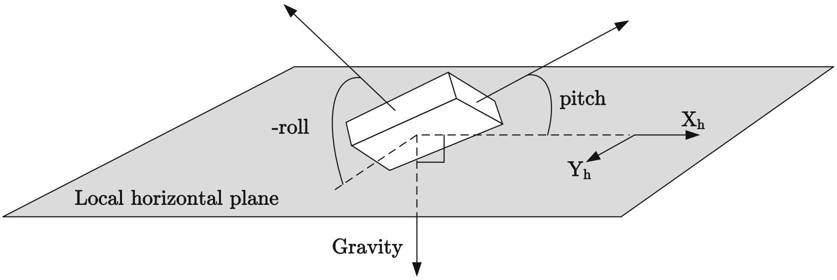


Fig. 10 Local horizontal plane as a reference

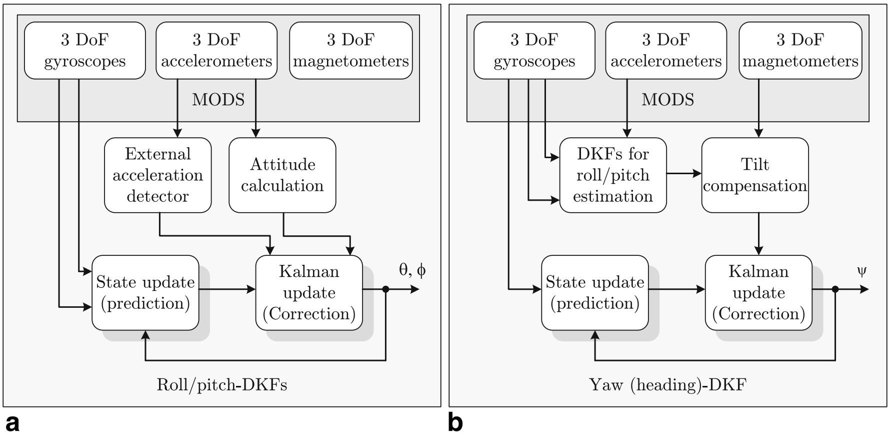


Fig. 11 **a** Discrete Kalman filter (DKF) for estimation of roll and pitch angles based on gyroscope and accelerometer measurements. **b** DKF for estimation of yaw (heading) angle from gyroscope and magnetometer measurements

Based on this approach a discrete Kalman filter bank (DKF-bank) is implemented which is responsible for the estimation of all three angles of the camera's orientation. For the pitch and the roll angle the same DKF-architecture is used, as indicated in Fig. 11a. In comparison the heading angle is estimated by an alternative architecture as shown in Fig. 11b.

The Kalman filtering process is composed from the following classical steps, where the following descriptions are simplified by referrring to just a single angle ξ.

Computation of an a priori state estimate $\mathbf{x}_{\mathbf{k+1}}^{-}$

As mentioned earlier the hidden states of the system are $\mathbf{x} = [\xi, \mathbf{b_{gyro}}]^{\mathbf{T}}$. The a priori estimates are computed by following the following relations:

$$\begin{aligned} \widehat{\omega}_{k+1} &= \omega_{k+1} - b_{gyro_k} \\ \xi_{k+1} &= \xi_k + \int \widehat{\omega}_{k+1} dt \\ b_{gyro_{k+1}} &= b_{gyro_k} \end{aligned} \tag{9}$$

Here the actual measurements from the gyroscopes ω_{k+1} are corrected for by the actually estimated bias b_{gyro_k} from the former iteration, before the actual angle ξ_{k+1} is computed.

Computation of a priori error covariance matrix $\mathbf{P^-_{k+1}}$

The a priori covariance matrix is calculated by incorporating the Jacobi matrix $\mathbf{A}$ of the states and the process noise covariance matrix $\mathbf{Q_K}$ as follows:

$$\mathbf{P^-_{k+1}} = \mathbf{A} \cdot \mathbf{P_k} \cdot \mathbf{A^T} + \mathbf{Q_K} \tag{10}$$

The two steps (1) and (2) are the elements of the prediction step as indicated in Fig. 11.

Computation of Kalman gain $\mathbf{K}_{k+1}$

As a prerequisite for computing the a posteriori state estimate the Kalman gain $\mathbf{K}_{k+1}$ has to be determined by following Eq. 11.

$$K_{k+1} = \mathbf{P^-_{k+1}} \cdot \mathbf{H}^T_{k+1} \cdot \left(\mathbf{H}_{k+1} \cdot \mathbf{P^-_{k+1}} \cdot \mathbf{H}^T_{k+1} + \mathbf{R}_{k+1}\right)^{-1} \tag{11}$$

Computation of a posteriori state estimate $\mathbf{x^+_{k+1}}$

The state estimate can now be corrected by using the calculated Kalman gain $\mathbf{K}_{k+1}$. Instead of incorporating the actual measurements as in the classical Kalman structure the suggested approach is based on the computation of an angle difference $\Delta\xi$. The difference is a comparison of the angle calculated from the gyroscope measures and the corresponding attitude as derived from the accelerometers, respectively the heading angle from the magnetometer, as already introduced in the introduction of this chapter. So the relation for $\mathbf{x^+_{k+1}}$ can be formulated as:

$$\mathbf{x^+_{k+1}} = \mathbf{x^-_{k+1}} - \mathbf{K}_{k+1} \cdot \Delta\xi \tag{12}$$

At this point it is important to consider the fact that the attitude measurements from the accelerometers are only reliable if there is no external translational motion. Thus an external acceleration detection is also needs to be part of the fusion procedure. For this reason the following condition (see Rehbinder et al. [12]) is evaluated continuously:

$$\|\mathbf{a}\| = \sqrt{(a_x^2 + a_y^2 + a_z^2)} \overset{!}{=} 1 \tag{13}$$

If the relation is fulfilled there is no external acceleration and the estimation of the attitude from accelerometers is more reliable than the one computed from rotational velocities as provided by the gyroscopes. For real sensors, a threshold $\varepsilon_{\mathbf{g}}$ is introduced to define an allowed variation from this ideal case. If the camera is not at rest the observation variance for the gyroscope data σ_g^2 is set to zero. By representing the magnitude of the acceleration measurements as $\|\mathbf{a}\|$ and the earth gravitational field $\mathbf{g} = [0, 0, -g]^T$ the observation variance can be defined by following Eq. 14.

$$\sigma_g^2 = \begin{cases} \sigma_g^2, & \|\mathbf{a}\| - \|\mathbf{g}\| < \varepsilon_{\mathbf{g}} \\ 0, & otherwise \end{cases} \tag{14}$$

A similar approach is chosen to overcome problems with the magnetometer measurements, in magnetically distorted environments for the DKF for the heading angle. The magnitude of the earth magnetic field **m** is evaluated as shown in the following Eq. 15[3], in an analogous way to Eq. 14 for describing variation due to gravity:

$$\sigma_g^2 = \begin{cases} \sigma_g^2, & \|\mathbf{m}\| - m_{des} < \varepsilon_{\mathbf{m}} \\ 0, & otherwise \end{cases} \tag{15}$$

Computation of posteriori error covariance matrix $\mathbf{P}^+_{\mathbf{k+1}}$

Finally the error covariance matrix is updated in the following way:

$$\mathbf{P}^+_{k+1} = \mathbf{P}^-_{k+1} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1} \cdot \mathbf{P}^-_{k+1} \tag{16}$$

It was shown in Aufderheide et al. [3], that the proposed strategy is able to outperform other classical algorithms for inertial sensor fusion, such as complementarity filtering or heuristic methods, in terms of accuracy and long-time stability.

If there is a robust estimate of the camera orientation available, it is possible to compute a 2D homography $\mathcal{H}$ which describes the optical flow (motion of all image pixels) between two successive image frames. According to Hwangbo et al. [7], it is possible to compute $\mathcal{H}$ for a pure rotational camera movement by using the following relation.

$$\mathcal{H}_k^{k+1} = \mathbf{K}\mathbf{R}_{CI}\mathbf{R}_k^{k+1}\mathbf{K}^{-1} \tag{17}$$

Here **K** represents the intrinsic camera parameters (such as focal length f, pixel size k, etc.), $\mathbf{R}_{CI}$ describes relative orientation between inertial and visual reference coordinate system and $\mathbf{R}_k^{k+1}$ describes the rotation of the camera between frame k and $k+1$, within the general frame-to-frame relative pose $\widetilde{\mathbf{M}}_k^{k+1}$.

2.6 Fusion for Position

At this point the orientation of the camera is known by following the classical strapdown approach. Hence, the position **p** can only be obtained by double integration of the body accelerations **a**, when a known orientation $\boldsymbol{\Xi} = [\phi\ \theta\ \psi]^T$ is available that allows a rotation from body frame B to reference (or navigation) frame N by using the direct cosine matrix (DCM) $\mathbf{C}_n^b$, defined as follows[4]:

$$\mathbf{C}_n^b = \begin{bmatrix} c\theta c\psi & s\varphi s\theta c\psi - c\varphi s\psi & c\varphi s\theta c\psi + s\varphi s\psi \\ c\theta s\psi & s\varphi s\theta s\psi + c\varphi c\psi & c\varphi s\theta s\psi - s\varphi c\psi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix} \tag{18}$$

[3] m_{des} describes the magnitude of the earth's magnetic field (e.g. 48 μT in Western Europe).

[4] For simplification: $s\alpha = sin(\alpha)$ and $c\beta = cos(\beta)$.

$$\mathbf{C}_n^b(\mathbf{q}) = \frac{1}{\sqrt{q_4^2 + \|\mathbf{e}\|^2}} \cdot \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2\left(q_1 q_2 + q_3 q_4\right) & 2\left(q_1 q_3 - q_2 q_4\right) \\ 2\left(q_1 q_2 - q_3 q_4\right) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2\left(q_2 q_3 + q_1 q_4\right) \\ 2\left(q_1 q_3 + q_2 q_4\right) & 2\left(q_2 q_3 - q_1 q_4\right) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{19}$$

The DCM can also be expressed in terms of an orientation quaternion $\mathbf{q} = [\mathbf{e}^T, q_4]^T$, where $\mathbf{e} = [q_1, q_2, q_3]^T$ describes the vector part and q_4 is the scalar part of $\mathbf{q}$. Equation 19 shows the relation between $\mathbf{C}_n^b$ and a computed $\mathbf{q}$. The actual position is computed by double integration of accelerometer measurements.

It should be noted here, that the absolute position estimate is affected by a much higher rate of uncertainty, because the double integration leads to an enormous drift which can not be bounded. The proposed approach for the visual-inertial feature tracking uses mainly frame-to-frame motion estimates, so that the drift within the absolute camera pose can be neglected.

3 Visual-Inertial Feature Tracking

Once there is a reliable motion estimate available it is very important to synchronise the inertial and the visual measurements. For this a basic clock signal is used to trigger both inertial sampling and acquiring images. The inertial measurements are available with a much higher frequency than the 30 frames per seconds (FPS) delivered by a standard camera module. Thus it is necessary to accumulate motion estimates from the S^3 to compute the frame-to-frame relative pose $\widetilde{\mathbf{M}}_k^{k+1}$.

Figure 12 shows the general architecture of the visual-inertial feature tracking system (VIFtrack!) for two subsequent frames of an image sequence.

The two camera positions for the frames $\mathbf{I_k}$ and $\mathbf{I_{k+1}}$ are related by a relative motion $\mathbf{M}_k^{k+1}$. The inertial smart sensor system is able to generate an estimate of that motion (translation and orientation) $\widetilde{\mathbf{M}}_k^{k+1}$ which can be used to update a set of parameters of the affine photometric motion model $\widehat{\mathbf{p}}_k^{k+1}$.

The chosen motion model should be able to compensate typical changes of the visual appearance of a descriptor over time. Here both photometric (illumination changes, etc.) and geometric changes of an image patch need to be considered. For this Jin et al. [8] propose a model which extended the classical affine geometric distortion proposed by Tomasi and Shi [15] by adding an photometric term.

The following equation shows the implementation of the model by using a parameter vector $\mathbf{p} = \left(\mathbf{A}_{[1,1]}, \mathbf{A}_{[1,2]}, \mathbf{A}_{[2,1]}, \mathbf{A}_{[2,2]}, \mathbf{d}_{[1]}, \mathbf{d}_{[2]}, \sigma, o\right)$ which contains the different elements of the affine warp ($\mathbf{A}$ and $\mathbf{d}$) and two photometric parameters (σ, o).

$$\Omega\left[\theta\left(\mathbf{I}_{k(\mathbf{x}_i)}\right)\right]_{\mathbf{p}} = (\sigma + 1)\,\theta\left(\mathbf{I}_k(\mathbf{A}\mathbf{x}_i + \mathbf{d})\right) + o \tag{20}$$

The photometric model is illustrated by Fig. 13, where a light source Λ illuminates a scene and the emitted light is reflected by the main surface S to the image plane Π, which is modelled by parameter σ.

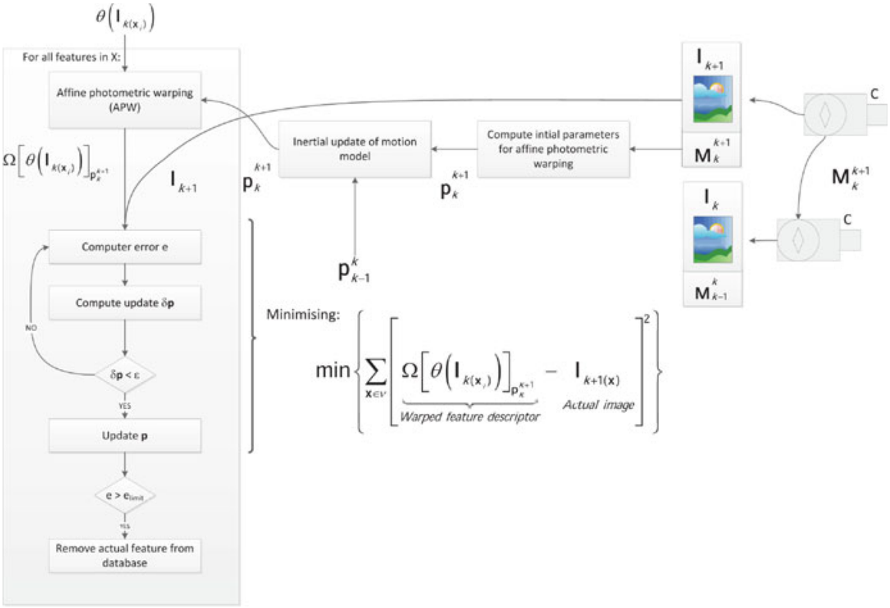

$\theta\left(\mathbf{I}_{k(\mathbf{x}_i)}\right)$
For all features in X:
Affine photometric warping (APW)
$\Omega\left[\theta\left(\mathbf{I}_{k(\mathbf{x}_i)}\right)\right]_{\mathbf{p}_k^{k+1}}$
$\mathbf{I}_{k+1}$
Computer error e
Compute update $\delta\mathbf{p}$
NO
$\delta\mathbf{p} < \varepsilon$
YES
Update $\mathbf{p}$
$e > e_{limit}$
YES
Remove actual feature from database
$\mathbf{p}_k^{k+1}$
Inertial update of motion model
Compute intial parameters for affine photometric warping
$\mathbf{p}_k^{k+1}$
$\mathbf{p}_{k-1}^{k}$
$\mathbf{I}_{k+1}$
$\mathbf{M}_k^{k+1}$
C
$\mathbf{M}_k^{k+1}$
$\mathbf{I}_k$
C
$\mathbf{M}_{k-1}^{k}$
Minimising:
$\min\left\{\sum_{\mathbf{x}\in\mathcal{V}}\left[\underbrace{\Omega\left[\theta\left(\mathbf{I}_{k(\mathbf{x}_i)}\right)\right]_{\mathbf{p}_k^{k+1}}}_{\text{Warped feature descriptor}} - \underbrace{\mathbf{I}_{k+1(\mathbf{x})}}_{\text{Actual image}}\right]^2\right\}$

Fig. 12 General scheme of the VIFtrack! approach

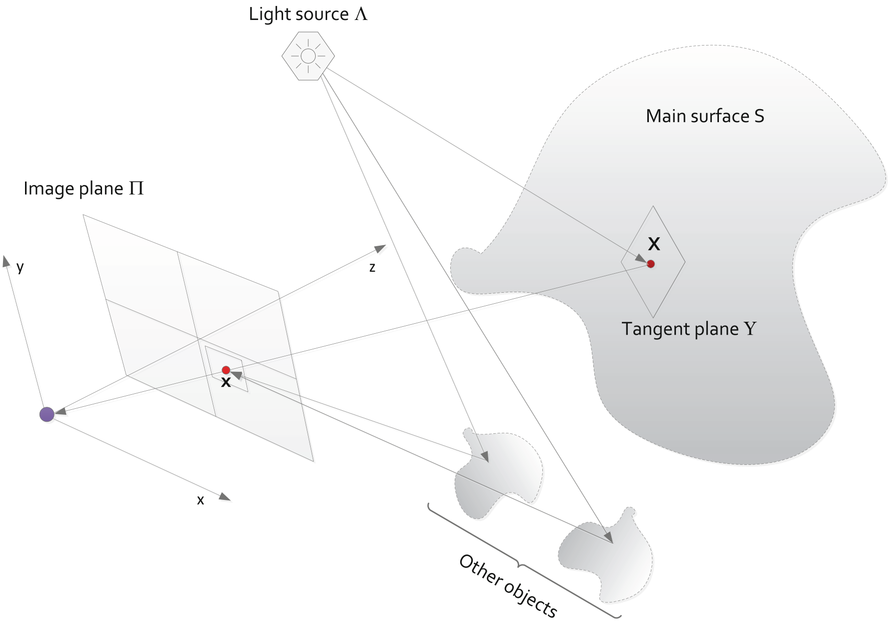


Fig. 13 Illustration of the photometric model with light rays reflected by the surface of the main object and reflectance from other objects

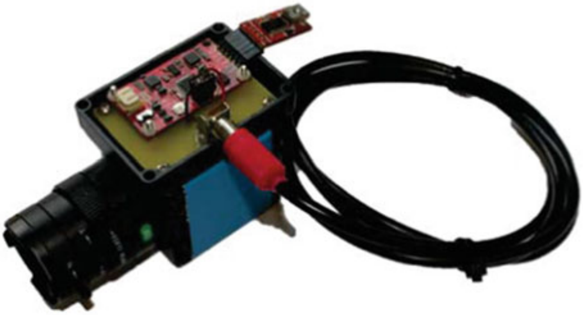


Fig. 14 Prototype of a visual-inertial sensor for VIFtrack!

Due to reflectance from other objects (ambient light sources) there are additional rays, which also change the intensity of an image pixel (parameter o). Due to the fact that the photometric motion cannot be estimated by using the inertial measurements, the corresponding values from the former frame are used as initial parameters for the optimisation. After the warping of the descriptors the optimisation process for each feature in X starts. For this optimisation, the following term needs to be minimized:

$$e = \min \left\{ \sum_{\mathbf{x} \in \nu} \left[\varOmega \left[\theta \left(\mathbf{I}_{k(\mathbf{x}_i)} \right) \right]_{\mathbf{p}_{k k+1}} - \mathbf{I}_{k+1(\mathbf{x})} \right]^2 \right\} \tag{21}$$

The minimisation problem can be approximated by a linearisation[5] around the actual set of parameters. Classical Gauss–Newton optimisation is used for finding the optimal set of parameters $\mathbf{p}$. As an abort criterion the actual change rate of $\mathbf{p}$ between two successive iterations is evaluated ($\delta \mathbf{p} < \varepsilon$).

The decision for determining whether a feature was successfully tracked can be made by evaluating the final value for e after the last iteration. If e lies above a certain threshold e_{limit} the feature is deleted from the feature database.

4 Results

The approach was evaluated by using a visual-inertial prototype (as shown in Fig. 14) which combines a standard industrial camera and the inertial smart sensor system. A microcontroller located on the S^3 is responsible for synchronising camera and IMU data.

An industrial robot was used in order to generate measurements with known motion, which can be used as ground truth sequences. Due to the fact that the background of the project is the area of 3D modelling, the used sequences contain only single

[5] For this a simple first-order Taylor expansion of the minimisation term is used.

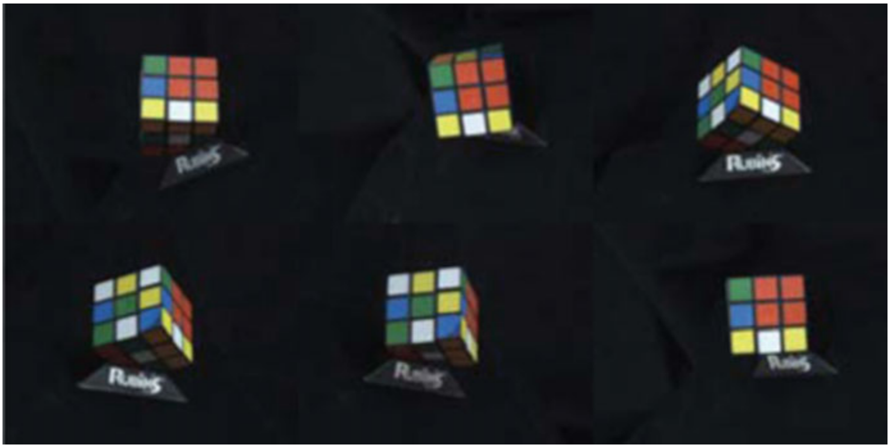


Fig. 15 Different frames of a test sequence "Object"

objects and a uniform background. The following figure illustrates exemplary frames of a typical sequence (Fig. 15).

We tested different motion patterns and optimised the corresponding parameters of the algorithm in order to produce best results. It was found that especially for high rotational velocities of the camera the VIFtrack! approach is able to outperform other feature tracking methods. Due to the fact that classical methods, such as the KLT-tracker from [11], utilise a purely translational model it is quite clear that especially a rolling camera leads to non-converging behaviour for many feature points. Figure 16 shows a typical motion pattern (slow camera speed) which we used for the evaluation. The suggested scheme can increase the number of successfully tracked features[6] up to 60 % in comparison to classical KLT for sequences with a rolling camera.

Figure 17 shows a comparison of the tracking performance for the VIFtrack!-method and the same principle (affine-photometric warping) only based on visual information for a given sequence. The mean number of successfully tracked features increases from 74 for visual-alone feature tracking up to 91 for the VIFtrack! scheme respectively. Especially for applications where a specific number of corresponding features is necessary (e.g. visual odometry) the VIFtrack!-method is useful, because while the visual-alone feature tracker loses up to 54 % of its feature points, VIFtrack! loses only up to 21 %.

The algorithm was also tested for a hand-held camera which was moved through an indoor environment. Figure 18 shows two typical examples for the tracking of features between two subsequent frames of the sequence. This sequence is more complex because the camera is freely moving within an indoor environment and no

[6] Here a successfully tracked feature is a feature which is not neglected based on the error threshold e_{limit}.

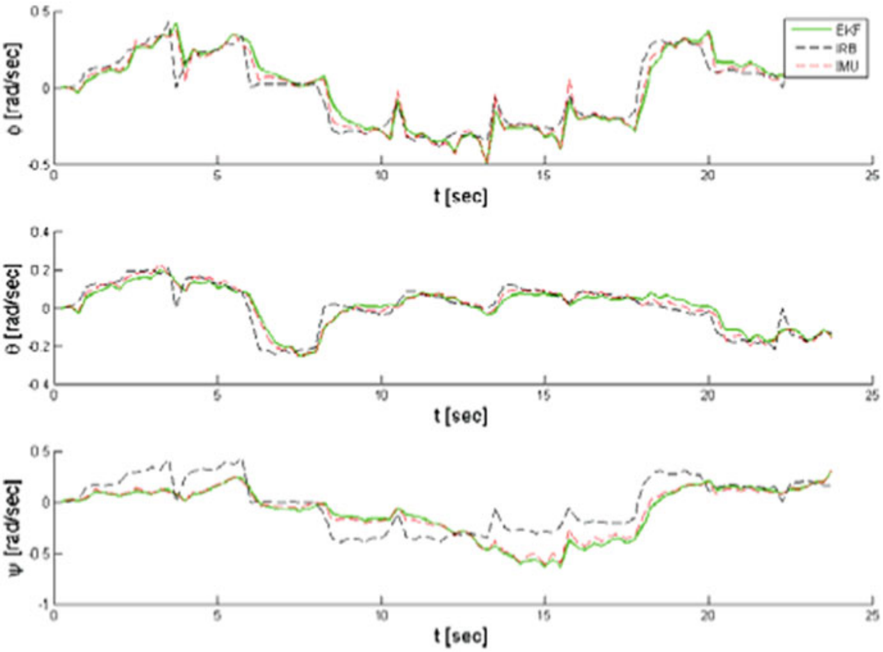


Fig. 16 Typical motion pattern for the evaluation describing rotation around the three Euler angles: *Black*: ground truth motion from industrial robot (IRB), *red*: measured angles from inertial measurements (IMU), *green*: estimated angles by fusion inertial and visual motion estimates (EKF))

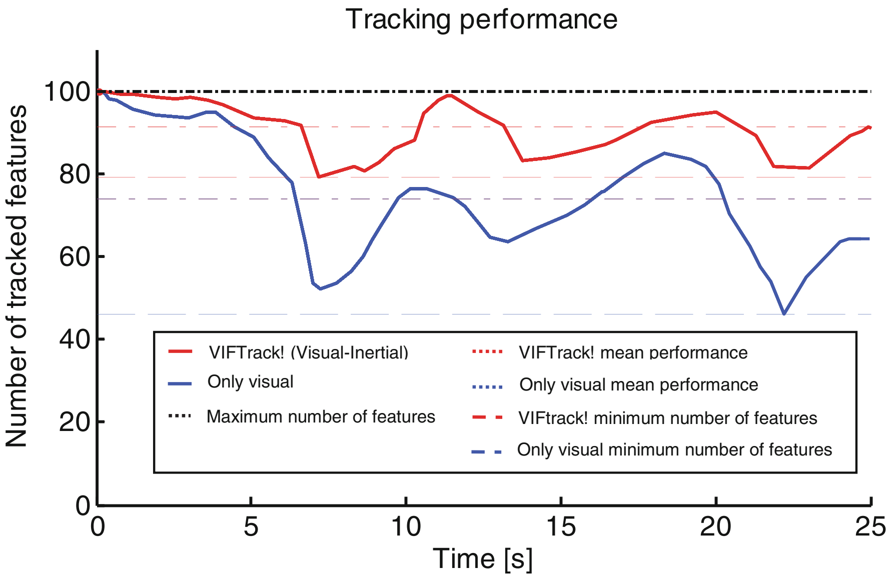


Fig. 17 Performance comparison between VIFtrack! and affine-photometric warping only based on visual information for the "object" sequence

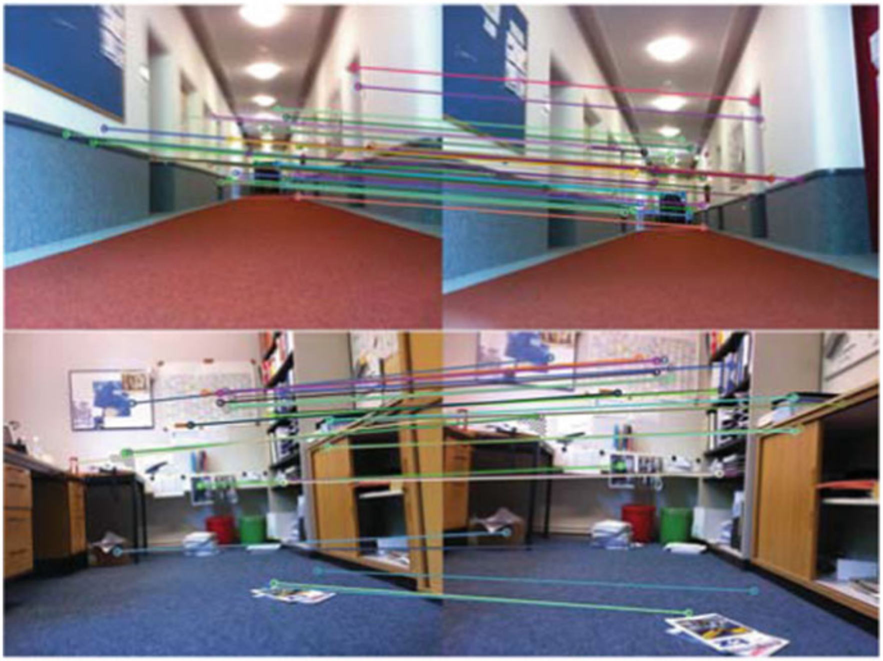


Fig. 18 Two examples for subsequent feature tracking results for the sequence gathered from a hand-held camera moved within an indoor environment

feature detected initially, within the first frame, remains visible for the entire sequence. For evaluating the VIFtrack! procedure a simple routine was introduced, which generates a set of feature candidates ${}^{1}X$ from the first frame. During the motion of the camera the number of successfully tracked features n decreases over time. Once n reaches a certain threshold ϖ, the algorithms generates a new set of feature candidates ${}^{k}X$ from the actual frame k of the sequence. This simple procedure should avoid that the tracking algorithm looses its track completely. The following table (Table 1) shows how often the algorithm generates a new set of feature candidates for the visual-inertial approach r_{VI} and classical KLT r_{KLT}.

Table 1 Comparison of the number of reinitialisation of feature candidates for VIFtrack! and classical KLT

n	r_{VI}	r_{KLT}	$\frac{r_{KLT}-r_{VI}}{r_{VI}}$ (%)
100	13	18	38
80	16	23	44
60	21	31	48
40	35	53	51
20	44	75	70

It can be seen from Table 1, that the usage of the VIFtrack! scheme is able to reduce the number of necessary re-initialisations of feature candidates due to the more robust feature tracking. Especially for a small number of initial feature candidates the visual-inertial feature tracking outperforms classical KLT.

5 Conclusion

The general problem of tracking a point feature throughout an image sequence acquired by a moving camera requires the implementation of an algorithm which is able to model the change of the visual appearance of each feature over time. The state of the art motion model used for feature tracking is an affine-photometric warping model, which models both changes in geometry and photometric conditions. For camera movements which involve high rotational velocities the 2D displacement of a point feature between two successive frames will increase dramatically. This leads to a non-converging behaviour of the minimisation problem, which adjusts a set of parameters in order to find the optimal match of the corresponding feature.

The usage of motion estimates, generated by an inertial smart sensor system as initial estimates for the motion model, leads to an increasing number of feature points, which can be successfully tracked throughout the whole sequence.

Future work will look into the possibility of fusing different motion estimates from visual and inertial cues, which would hopefully lead to a higher robustness against incorrect inertial measurements. For this visual-based relative pose estimators need to be evaluated to get a handle on the accuracy (see Aufderheide et al. [4]).

References

1. Aufderheide D, Krybus W (2010) Towards real-time camera egomotion estimation and three-dimensional scene acquisition from monocular image streams. In: Proceedings of the 2010 international conference on Indoor Positioning and Indoor Navigation (IPIN 2010). Zurich, Switzerland, September, 15–17 2010, pp 1–10. IEEE – ISBN 978-1-4244-5862-2
2. Aufderheide D, Steffens M, Kieneke S, Krybus W, Kohring C, Morton D (2009) Detection of salient regions for stereo matching by a probabilistic scene analysis. In: Proceedings of the 9th conference on optical 3-D measurement techniques. Vienna, Austria, July, 1–3 2009, pp 328–331. ISBN 978-3-9501492-5-8
3. Aufderheide D, Krybus W, Dodds D (2011) A MEMS-based smart sensor system for estimation of camera pose for computer vision applications. In: Proceedings of the University of Bolton Research and Innovation Conference 2011, Bolton, U.K., June, 28–29 2011, The University of Bolton Institutional Repository
4. Aufderheide D, Krybus W, Witkowski U, Edwards G (2012) Solving the PnP problem for visual odometry—an evaluation of methodologies for mobile robots. In: Advances in autonomous robotics—joint proceedings of the 13th annual TAROS conference and the 15th annual FIRA RoboWorld Congress Bristol, UK, August 20–23, pp 461–462
5. Harris C, Stephens M (1988) A combined corner and edge detector. In: Proceedings of the 4th Alvey vision conference, pp 147–151

6. Hwangbo M, Kim JS, Kanade T (2009) Inertial-aided KLT feature tracking for a moving camera. In: 2009 IEEE/RJS international conference on intelligent robots and systems. St. Louis, USA, pp 1909–1916
7. Hwangbo M, Kim JS, Kanade T (2011) Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and GPU implementation. Int J Robot Res 30(14):1755–1774
8. Jin H, Favaro P, Soatto S (2001) Real-time feature tracking and outlier rejection with changes in illumination. In: Proceedings of the International Conference on Computer Vision (ICCV), July 2001
9. Juan L, Gwun O (2009) A comparison of SIFT, PCA-SIFT and SURF. Int J Image Process (IJIP) 3(4):143–152. CSC Journals
10. Kim J, Hwangbo M, Kanade T (2009) Realtime affine-photometric KLT feature tracker on GPU in CUDA framework. The fifth IEEE workshop on embedded computer vision in ICCV 2009, Sept 2009, pp 1306–1311
11. Lucas B, Kanade T (1981) An iterative image registration technique with an application to Stereo vision. In: International joint conference on artificial intelligence, pp 674–679
12. Rehbinder H, Hu X (2004) Drift-free attitude estimation for accelerated rigid bodies. Automatica 40(4):653–659
13. Sabatini A (2006) Quaternion-based extended Kalman filter for determining orientations by inertial and magnetic sensing. IEEE Trans Biomed Eng 53(7):1346–1356
14. Skog I, Haendel P (2006) Calibration of MEMS inertial unit. In: Proceedings of the IXVII IMEKO world congress on metrology for a sustainable development
15. Tomasi C, Shi J (1994) Good features to track. In: IEEE computer vision and pattern recognition 1994
16. Welch G, Bishop G (2006) An introduction to the Kalman filter, Technical Report TR 95-041. Department of Computer Science, University of North Carolina at Chapel Hill