J Real-Time Image Proc (2007) 2:149–160
DOI 10.1007/s11554-007-0040-2

SPECIAL ISSUE

Robust real-time tracking by fusing measurements from inertial and vision sensors

Jeroen D. Hol · Thomas B. Schön · Henk Luinge · Per J. Slycke · Fredrik Gustafsson

Received: 4 June 2007 / Accepted: 23 August 2007 / Published online: 18 October 2007

Abstract The problem of estimating and predicting position and orientation (pose) of a camera is approached by fusing measurements from inertial sensors (accelerometers and rate gyroscopes) and vision. The sensor fusion approach described in this contribution is based on nonlinear filtering of these complementary sensors. This way, accurate and robust pose estimates are available for the primary purpose of augmented reality applications, but with the secondary effect of reducing computation time and improving the performance in vision processing. A real-time implementation of a multi-rate extended Kalman filter is described, using a dynamic model with 22 states, where 12.5 Hz correspondences from vision and 100 Hz inertial measurements are processed. An example where an industrial robot is used to move the sensor unit is presented. The advantage with this configuration is that it provides ground truth for the pose, allowing for objective performance evaluation. The results show that we obtain an absolute accuracy of 2 cm in position and 1° in orientation.

J. D. Hol (✉) · T. B. Schön · F. Gustafsson
Division of Automatic Control,
Linköping University, 581 83 Linköping, Sweden
e-mail: hol@isy.liu.se

T. B. Schön
e-mail: schon@isy.liu.se

F. Gustafsson
e-mail: fredrik@isy.liu.se

H. Luinge · P. J. Slycke
Xsens Technologies B.V.,
Pantheon 6a, Postbus 559,
7500 AN Enschede, The Netherlands
e-mail: henk@xsens.com

P. J. Slycke
e-mail: per@xsens.com

Keywords Pose estimation · Sensor fusion · Computer vision · Inertial navigation

1 Introduction

This paper deals with estimating the position and orientation (pose) of a camera in real-time, using measurements from inertial sensors (accelerometers and rate gyroscopes) and a camera. A system has been developed to solve this problem in unprepared environments, assuming that a map or a scene model is available. For a more detailed description of the overall system and the construction of scene models we refer to Stricker and Thomas [37] and Bartczak et al. [24], respectively. In this paper, the sensor fusion part of the system is described, which is based upon a rather general framework for nonlinear state estimation available from the statistical signal processing community.

This problem under ideal conditions can be solved using only a camera. Hence, it might seem superfluous to introduce inertial sensors. However, the most important reasons justifying an *inertial measurement unit* (IMU) are:

- Producing more *robust* estimates. Any single camera system will experience problems during periods with uninformative or no vision data. This will occur, typically due to occlusion or fast motion. An IMU will help to bridge such gaps, which will be illustrated in the present paper.
- Reducing computational demands for image processing. Accurate short time pose estimates are available using the information from the IMU, reducing the need for fast vision updates.

The combination of vision and inertial sensors has been used previously in literature. Corke et al. [7] gave an introduction to this field and its applications. Reported systems apply various methods: inertial measurements are used as backup [2], for short time pose prediction [23], or depth map alignment [26]. Alternatively, vision and inertial subsystems are loosely coupled, using visual pose measurements [1, 6, 31]. Vision relies on either specific targets, line contours or natural landmarks. Calibration of the sensors is discussed in e.g., Lobo and Dias [27]. Furthermore, the problem is closely related to the problem of *simultaneous localization and mapping* (SLAM) [11, 39], where camera tracking and scene model construction are performed simultaneously. Single camera SLAM is discussed in Davison [8] and Davison et al. [10]. In that context so-called fast localization algorithms [41] are investigated as alternatives to inertial support [13, 30].

In our approach, real-time camera pose estimation is achieved by fusing inertial and vision measurements using the framework of nonlinear state estimation, covering methods such as the *Extended Kalman Filter* (EKF), the *Unscented Kalman Filters* (UKF) and the *particle filter* (PF). This results in a tightly coupled system, naturally supporting multi-rate signals. The vision measurements are based on natural landmarks, which are detected guided by pose predictions. The measurements from the sensors are used directly rather than being processed to a vision-based pose or an inertial-based pose. The components of the system are well known. However, we believe that the way in which these components are assembled is novel and we show that the resulting system provides accurate and robust pose estimates.

The sensors generating the measurements y_t are described in Sect. 2. In Sect. 3, the framework for state estimation in nonlinear dynamic systems is introduced in more detail and used to solve the sensor fusion problem we are faced with in the present application. In implementing this, there are several practical issues that have to be solved. The overall performance of the system heavily relies on successful solutions to these matters, which is explained in Sect. 4. The performance of the implementation is evaluated in Sect. 5, and finally, the paper is concluded in Sect. 6.

2 Sensors

An IMU and a digital video camera are combined to provide measurements to the sensor fusion module, described in this paper. Both sensors are relatively small and unobtrusive and they can be conveniently integrated into a single *sensor unit*. An example of a prototype is shown in Fig. 1. An on board digital signal processor containing calibration parameters is used to calibrate and synchronize data from the different components.

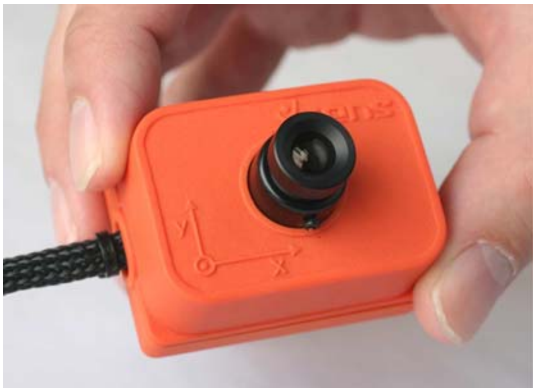

Fig. 1 A prototype of the MATRIS project, integrating a camera and an IMU in a single housing. It provides a hardware synchronized stream of video and inertial data

Before discussing the inertial and vision sensors in the subsequent sections, the required coordinate systems are introduced.

2.1 Coordinate systems

When working with a sensor unit containing a camera and an IMU several coordinate systems have to be introduced:

- ***Earth (e)***: The camera pose is estimated with respect to this coordinate system. It is fixed to earth and the features of the scene are modelled in this coordinate system. It can be aligned in any way; however, preferably it should be vertically aligned.
- ***Camera (c)***: The coordinate system is attached to the moving camera. Its origin is located in the optical center of the camera, with the z-axis pointing along the optical axis. The camera, a projective device, acquires its images in the ***image plane (i)***. This plane is perpendicular to the optical axis and is located at an offset (focal length) from the optical center of the camera.
- ***Body (b)***: This is the coordinate system of the IMU. Even though the camera and the IMU are rigidly attached to each other and contained within a single package, the body coordinate system does not coincide with the camera coordinate system. They are separated by a constant translation and rotation.

These coordinate systems are used to denote geometric quantities, for instance, $\boldsymbol{c}^{\mathrm{e}}$ is the position of the camera coordinate system expressed in the earth system and R^{cb} is the rotation matrix from the body system to the camera system.

2.2 Inertial sensors

The sensor unit contains an IMU with three perpendicularly mounted 1,200 °/s ADXLXRS300 angular velocity sensors and two 5 g 2D ADXL22293 accelerometers, which are mounted such that three of the sensitive axes are perpendicular to each other. MEMS rate gyroscopes are chosen because of their dramatically reduced size and low cost as compared to alternatives such as fiber optic angular velocity sensors.

The signals from the inertial components are synchronously measured at 100 Hz using a 16 bit A/D converter. A temperature sensor is added to compensate for the temperature dependency of the different sensing components.

The assembly containing the gyroscopes and accelerometers has been subjected to a calibration procedure to calibrate for the exact physical alignment of each component, the gains, the offsets and the temperature relations of the gains and offsets. With these a 3D angular velocity vector and a 3D accelerometer vector, both resolved in the body coordinate system, are computed using an on board processor. See e.g., Titterton and Weston [40] and Chatfield [5] for suitable background material on inertial sensors and the associated signal processing.

The calibrated gyroscope signal $\boldsymbol{y}_{\omega,t}$ contains measurements of the angular velocity $\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{eb},t}$ from body to earth (eb) expressed in the body coordinate system (b):

$$\boldsymbol{y}_{\omega,t} = \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{eb},t} + \boldsymbol{\delta}^{\mathrm{b}}_{\omega,t} + \boldsymbol{e}^{\mathrm{b}}_{\omega,t}. \tag{1}$$

Even though the gyroscope signal is corrected for temperature effects, some low-frequency offset fluctuations $\boldsymbol{\delta}_{\omega,t}$ remain, partly due to the unmodeled acceleration dependency. The remaining error $\boldsymbol{e}^{\mathrm{b}}_{\omega,t}$ is assumed to be zero mean white noise. The measurements are not accurate enough to pick up the rotation of the earth. This implies that the earth coordinate system can be considered to be an inertial frame.

A change in orientation can be obtained by proper integration of the gyroscope signal. This orientation can be obtained even during fast and abrupt movements, not relying on any infrastructure other than the gyroscope itself. However, the accuracy in orientation will deteriorate for periods longer than a few seconds.

The calibrated accelerometer signal $\boldsymbol{y}_{\mathrm{a},t}$ contains measurements of the combination of the body acceleration vector $\ddot{\boldsymbol{b}}_t$ and the gravity vector $\boldsymbol{g}$, both expressed in the body coordinate system:

$$\boldsymbol{y}_{\mathrm{a},t} = \ddot{\boldsymbol{b}}^{\mathrm{b}}_t - \boldsymbol{g}^{\mathrm{b}} + \boldsymbol{\delta}^{\mathrm{b}}_{\mathrm{a},t} + \boldsymbol{e}^{\mathrm{b}}_{\mathrm{a},t}. \tag{2}$$

Even though the accelerometer measurement is corrected for temperature effects a small low-frequency offset $\boldsymbol{\delta}_{\mathrm{a},t}$ remains. The error $\boldsymbol{e}^{\mathrm{b}}_{\mathrm{a},t}$ is assumed to be zero mean white noise.

Gravity is a constant vector in the earth coordinate system. However, expressed in body coordinates gravity depends on the orientation of the sensor unit. This means that once the orientation is known, the accelerometer signal can be used to estimate the acceleration, or alternatively, once the acceleration is known, the direction of the vertical can be estimated.

Accelerations can be integrated twice to obtain a change in position. This can be done during fast and abrupt motions as long as an accurate orientation estimate is available, for instance from the gyroscopes. However, the accuracy of the position change will deteriorate quickly as a result of the double integration and the sensitivity with respect to orientation errors.

2.3 Monocular vision

Apart from the inertial sensors, the sensor unit is equipped with a ptGrey DragonFly CCD camera with a perspective lens with a focal length of 3.2 mm. Color images with a resolution of 320 × 240 pixels at a frame rate of 12.5 Hz are streamed to a PC using a firewire connection. The camera is triggered by the IMU clock allowing for synchronized measurements.

This setup is one realization of monocular vision: cameras can vary in sensor type, resolution, frame rate, and various lens types can be used ranging from perspective to fish-eye. However, they remain projective devices, that is, they are bearings only sensors which do not provide distance directly.

Extracting camera position and orientation from images is a known and well studied problem in computer vision [28, 16]. The key ingredient is to find correspondences, relations between a feature found in the image which corresponds to an element in the scene model. All these are rather abstract concepts, which do have numerous implementations, ranging from Harris detectors [15] and point clouds models to patches and textured free-form surfaces models [24]. The correspondences are the pieces of information which can be extracted from an image, and they will be considered to be the vision measurements in this article.

Point correspondences $\boldsymbol{z}^c \leftrightarrow \boldsymbol{z}^i$ are the relation between 3D points $\boldsymbol{z}^c$ and 2D image points $\boldsymbol{z}^i$. For a perspective lens and a pinhole camera the correspondence relation is

$$\boldsymbol{z}^i = \begin{pmatrix} f z^c_x / z^c_z \\ f z^c_y / z^c_z \end{pmatrix} + \boldsymbol{e}^i, \tag{3a}$$

or equivalently,

$$\boldsymbol{0} \approx \begin{pmatrix} -fI_2 & z_t^i \end{pmatrix} z_t^c = \begin{pmatrix} -fI_2 & z_t^i \end{pmatrix} R_t^{\mathrm{ce}} \left(z^e - \boldsymbol{c}_t^e \right), \tag{3b}$$

where f is the focal length and I_2 the 2 × 2 identity matrix. The error $\boldsymbol{e}_t^i$ is assumed to be a zero mean white noise. Here, it is worth noting that this assumption is not that realistic, due to outliers, quantization effects etc. From (3b) it can be seen that the camera pose depends on the rotation matrix R^{ce} and the position $\boldsymbol{c}^{\mathrm{e}}$. Hence, given sufficient correspondences and a calibrated camera the camera pose can be solved. Similar relations can be derived for e.g., line correspondences which also provide information about the camera pose and optical velocity fields which provide information about the camera velocity [7].

Correspondences are bearings only measurements and as such they provide information about absolute position and orientation with respect to the earth coordinate system. Note that everything is determined up to a scale ambiguity; viewing a twice as large scene from double distance will yield an identical image. However, these vision measurements are available at a relatively low rate due to the trade off between exposure time and accuracy (pixel noise and motion blur) which is an important limit for small aperture cameras. Furthermore, processing capacity might constrain the frame rate. Hence, the observed image can change drastically from frame to frame, which occurs already with normal human motion. This is the main cause for the limited robustness inherent in single camera systems.

The computer vision implementation used in the present implementation is based on a *sum of absolute difference* (SAD) block matcher in combination with a planar patch or free-form surface model of the scene. More details can be found in Stricker and Thomas [37], Bartczak et al. [24] and Skoglund and Felsberg [35]. Both pixel data and 3D positions are stored for each feature. An example of a scene model is shown in Fig. 2. While tracking, search templates are generated by warping the patches in the model according to homographies calculated from the latest prediction of the camera pose. These templates are then matched with the current calibrated camera image using the block matcher. In this way correspondences are generated.

3 Sensor fusion

The inertial and vision sensors contained in the sensor unit have complementary properties. Vision in combination with the map gives accurate absolute pose information at a low rate, but experiences problems during moderately fast motions. The IMU provides high rate relative pose information regardless of the motion speed, but becomes inaccurate after a short period of time. By fusing information from both sources it is possible to obtain robust camera pose estimates.

Combing inertial and vision sensors is possible in several ways. For instance, vision-based methods might be extended by using pose predictions from the IMU. These pose predictions can be used to determine where in the image the features are to be expected. Once detected, the features can be used to calculate the pose and this pose is then used as a starting point for the next pose prediction by the IMU. Alternatively, the IMU can be considered to be the main sensor, which is quite common in the navigation industry. In that case, vision can be used for error correction, similar to how radio beacons or the *global positioning system* (GPS) are used to correct the drift in an *inertial navigation system* (INS).

Although the sensors have different properties, it is from a signal processing perspective not relevant to assign a ‘main’ sensor and an ‘aiding’ sensor. Both vision and inertial sensors are equivalent in the sense that they both provide information about the quantity of interest, the camera pose in this application. The objective is to extract as much information as possible from the measurements. More specifically, this amounts to finding the best possible estimate of the filtering *probability density function* (pdf) $p(x_t|y_{1:t})$, where $y_{1:t} \triangleq \{y_1, \ldots, y_t\}$. The topic of this section

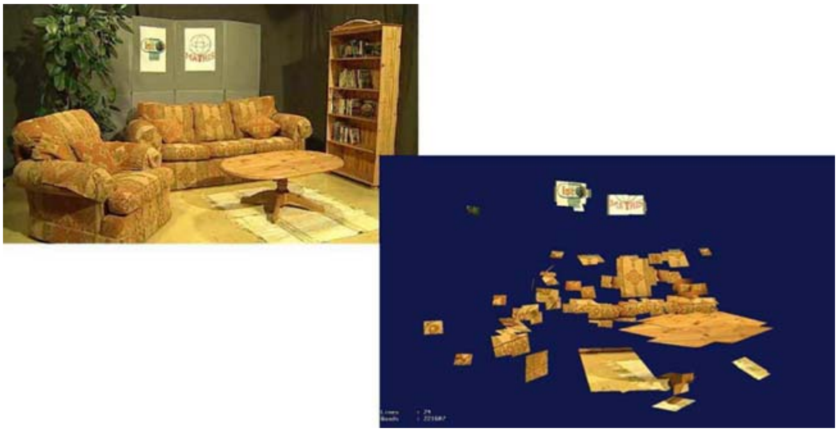

Fig. 2 An example of a scene model consisting of planar patches (*lower right*) and the actual scene that is modelled (*upper left*)

is to provide a solid framework for computing approximations of this type. First, a rather general introduction to this framework is given in Sect. 3.1. The rest of this section is devoted to explaining how this framework can be applied to handle the present application. The models are introduced in Sect. 3.2 and the fusion algorithm is discussed in Sect. 3.3.

3.1 Theoretical framework

The objective in sensor fusion is to recursively in time estimate the state in the dynamic model,

$$x_{t+1} = f_t(x_t, u_t, v_t), \tag{4a}$$

$$y_t = h_t(x_t, u_t, e_t), \tag{4b}$$

where $x_t \in \mathbb{R}^{n_x}$ denotes the state, $y_t \in \mathbb{R}^{n_y}$ denote the measurements from a set of sensors, v_t and e_t denote the stochastic process and measurement noise, respectively. The process model equations, describing the evolution of the states (pose etc.) over time are denoted by $f: \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$. Furthermore, the measurement model is given by $h: \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_e} \to \mathbb{R}^{n_y}$, describing how the measurements from the IMU and the camera relate to the state. The goal is to infer all the information from the measurements y_t onto the state x_t. The way of doing this is to compute the filtering pdf $p(x_t|y_{1:t})$. The filtering pdf contains everything there is to know about the state at time t, given the information in all the past measurements $y_{1:t}$. Once an approximation of $p(x_t|y_{1:t})$ is available it can be used to form many different (point) estimates, including maximum likelihood estimates, confidence intervals and the most common conditional expectation estimate

$$\hat{x}_t = E(x_t|y_{1:t}). \tag{5}$$

The key element in solving the nonlinear state estimation problem in real time is the propagation of $p(x_t|y_{1:t})$ over time. It is well known, see [19] that a recursive solution can be obtained by applying Bayes' theorem, introducing model (4) in the iterations,

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})\mathrm{d}x_t}, \tag{6a}$$

$$p(x_{t+1}|y_{1:t}) = \int p(x_{t+1}|x_t)p(x_t|y_{1:t})\mathrm{d}x_t. \tag{6b}$$

Hence, the quality of the solution is inherently coupled to the models and hence good models are imperative. It is worth noticing that (6a) and (6b) are often referred to as measurement update and time update, respectively. The sensor fusion problem has now been reduced to propagating (6) over time as new measurements arrive. The problem is that the multidimensional integrals present in (6) lack analytical solutions in all but a few special cases. The most common special case is when (4) is restricted to be a linear dynamic system, subject to additive Gaussian noise. Then all the involved densities will be Gaussian, implying that it is sufficient to propagate the mean and covariance. The recursions updating these are of course given by the Kalman filter [21].

However, in most cases there does not exist a closed form solution for (6), forcing the use of approximations of some sort. The literature is full of different ideas on how to perform these approximations. The most common being the EKF [36, 32] where the model is linearized and the standard Kalman filter equations are used for this linearized model. A conceptually more appealing approximation is provided by the PF [14, 18, 22] which retains the model and approximates (6). Other popular approximations for the nonlinear state estimation problem are provided for example by the UKF [20] and the point-mass filter [3, 4]. For a more complete account of the nonlinear state estimation problem, see e.g., Schön [33].

3.2 Models

The probability density functions $p(x_{t+1}|x_t)$ and $p(y_t|x_t)$ are the key elements in the filter iterations (6). They are usually implicitly specified by the process model (4a) and the measurement model (4b). For most applications the model formulation given in (4) is too general. It is often sufficient to assume that the noise enters additively, according to

$$x_{t+1} = f_t(x_t) + v_t, \tag{7a}$$

$$y_t = h_t(x_t) + e_t. \tag{7b}$$

The fact that the noise is additive in (7) allows for explicit expressions for $p(x_{t+1}|x_t)$ and $p(y_t|x_t)$, according to

$$p(x_{t+1}|x_t) = p_{v_t}(x_{t+1} - f_t(x_t)), \tag{8a}$$

$$p(y_t|x_t) = p_{e_t}(y_t - h_t(x_t)), \tag{8b}$$

where $p_{v_t}(\cdot)$ and $p_{e_t}(\cdot)$ denote the pdf's for the noise v_t and e_t, respectively. Note that the input signal u_t has been dispensed with, since it does not exist in the present application. The rest of this section will discuss the model used in the current application.

First of all, the state vector has to include the position and the orientation, since they are the quantities of interest. However, in order to be able to use the IMU and provide predictions the state vector should also include their time derivatives, as well as sensor biases. The state vector is chosen to be

$$x_t = \left(\boldsymbol{b}_t^{\mathrm{e}}\ \dot{\boldsymbol{b}}_t^{\mathrm{e}}\ \ddot{\boldsymbol{b}}_t^{\mathrm{e}}\ q_t^{\mathrm{be}}\ \boldsymbol{\omega}_{\mathrm{eb},t}^{\mathrm{b}}\ \boldsymbol{\delta}_{\omega,t}^{\mathrm{b}}\ \boldsymbol{\delta}_{\mathrm{a},t}^{\mathrm{b}}\right)^{\mathrm{T}}. \tag{9}$$

That is, the state vector consists of position of the IMU (the body coordinate system) expressed in the earth system $\boldsymbol{b}^{\mathrm{e}}$, its velocity $\dot{\boldsymbol{b}}^{\mathrm{e}}$ and acceleration $\ddot{\boldsymbol{b}}^{\mathrm{e}}$, the orientation of the body with respect to the earth system q^{be}, its angular velocity $\boldsymbol{\omega}_{\mathrm{eb}}^{\mathrm{b}}$, the gyroscope bias $\boldsymbol{\delta}_{\omega}^{\mathrm{b}}$ and the accelerometer bias $\boldsymbol{\delta}_{\mathrm{a}}^{\mathrm{b}}$. All quantities are three dimensional vectors, except for the orientation which is described using a four-dimensional unit quaternion q^{be}, resulting in a total state dimension of 22. Parameterization of a three-dimensional orientation is in fact rather involved, see e.g., Shuster [34] for a good account of several of the existing alternatives. The reason for using unit quaternions is that they offer a nonsingular parameterization with a rather simple dynamics. Using (9) as state vector, the process model is given by

$$\boldsymbol{b}_{t+1}^{\mathrm{e}} = \boldsymbol{b}_t^{\mathrm{e}} + T\dot{\boldsymbol{b}}_t^{\mathrm{e}} + \frac{T^2}{2}\ddot{\boldsymbol{b}}_t^{\mathrm{e}}, \tag{10a}$$

$$\dot{\boldsymbol{b}}_{t+1}^{\mathrm{e}} = \dot{\boldsymbol{b}}_t^{\mathrm{e}} + T\ddot{\boldsymbol{b}}_t^{\mathrm{e}}, \tag{10b}$$

$$\ddot{\boldsymbol{b}}_{t+1}^{\mathrm{e}} = \ddot{\boldsymbol{b}}_t^{\mathrm{e}} + \boldsymbol{v}_{\ddot{b},t}^{\mathrm{e}}, \tag{10c}$$

$$q_{t+1}^{\mathrm{be}} = \exp\left(-\frac{T}{2}\boldsymbol{\omega}_{\mathrm{eb},t}^{\mathrm{b}}\right) \odot q_t^{\mathrm{be}}, \tag{10d}$$

$$\boldsymbol{\omega}_{\mathrm{eb},t+1}^{\mathrm{b}} = \boldsymbol{\omega}_{\mathrm{eb},t}^{\mathrm{b}} + \boldsymbol{v}_{\omega,t}^{\mathrm{b}}, \tag{10e}$$

$$\boldsymbol{\delta}_{\omega,t+1}^{\mathrm{b}} = \boldsymbol{\delta}_{\omega,t}^{\mathrm{b}} + \boldsymbol{v}_{\delta_\omega,t}^{\mathrm{b}}, \tag{10f}$$

$$\boldsymbol{\delta}_{\mathrm{a},t+1}^{\mathrm{b}} = \boldsymbol{\delta}_{\mathrm{a},t}^{\mathrm{b}} + \boldsymbol{v}_{\delta_{\mathrm{a}},t}^{\mathrm{b}}, \tag{10g}$$

where the quaternion multiplication and exponential are defined according to

$$\begin{pmatrix} p_0 \\ \boldsymbol{p} \end{pmatrix} \odot \begin{pmatrix} q_0 \\ \boldsymbol{q} \end{pmatrix} \triangleq \begin{pmatrix} p_0 q_0 - \boldsymbol{p}\cdot\boldsymbol{q} \\ p_0\boldsymbol{q} + q_0\boldsymbol{p} + \boldsymbol{p}\times\boldsymbol{q} \end{pmatrix}, \tag{11a}$$

$$\exp(\boldsymbol{v}) \triangleq \begin{pmatrix} \cos\|\boldsymbol{v}\| \\ \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\sin\|\boldsymbol{v}\| \end{pmatrix}. \tag{11b}$$

A standard constant acceleration model (10a)–(10c) has been used to model the position, velocity and acceleration. Furthermore, the quaternion dynamics is standard, see e.g., Shuster [34]. Finally, the angular velocity and the bias terms are simply modeled as random walks, since there is no systematic knowledge available about these terms.

There is more than one sensor type available, implying that several measurement models are required. They have already been introduced in Sect. 2, but for convenience they are all collected here,

$$\boldsymbol{y}_{\mathrm{a},t} = R_t^{\mathrm{be}}(\ddot{\boldsymbol{b}}_t^{\mathrm{e}} - \boldsymbol{g}^{\mathrm{e}}) + \boldsymbol{\delta}_{\mathrm{a},t}^{\mathrm{b}} + \boldsymbol{e}_{\mathrm{a},t}^{\mathrm{b}}, \tag{12a}$$

$$\boldsymbol{y}_{\omega,t} = \boldsymbol{\omega}_{\mathrm{eb},t}^{\mathrm{b}} + \boldsymbol{\delta}_{\omega,t}^{\mathrm{b}} + \boldsymbol{e}_{\omega,t}^{\mathrm{b}}, \tag{12b}$$

$$\boldsymbol{y}_{\mathrm{c},t} = \left(-fI_2 \quad \boldsymbol{z}_t^i\right) R^{\mathrm{cb}}\left(R_t^{\mathrm{be}}\left(\boldsymbol{z}_t^{\mathrm{e}} - \boldsymbol{b}_t^{\mathrm{e}}\right) - \boldsymbol{c}_t^{\mathrm{b}}\right) + \boldsymbol{e}_{\mathrm{c},t}. \tag{12c}$$

Note that the rotation matrix R_t^{be} is constructed from q_t^{be} [25]. The transformation from body to camera coordinate system is included in (12c), compared to (3b).

3.3 Fusion algorithm

The nonlinear estimation framework discussed in Sect. 3.1 suggests Algorithm 1 to fuse the multi-rate information from the inertial and vision sensors. The algorithm uses the models (10) and (12) to perform the time and measurement update steps given in (6). Note that Algorithm 1 is generic in the sense that we have not specified which state estimation algorithm is used. Our implementation, which runs in real time with 100 Hz inertial measurements and frame rates up to 25 Hz, uses the EKF to compute the estimates, implying that all involved pdf's are approximated by Gaussian densities. An UKF implementation was found to give similar accuracy at the cost of a higher computational burden [29]. This confirms the results from Armesto et al. [1].

When the sensor unit is static during initialization, the IMU provides partial or full (using magnetometers)

Algorithm 1 Recursive camera pose calculation

```
1. Perform an initialization and set initial state estimate and
   covariance.
     $x_0 \sim p(x_o)$
2. Time update. Calculate $p(x_t|y_{1:t-1})$ by propagating $p(x_{t-1}|y_{1:t-1})$
   through the process model (10).
3. Accelerometer and gyroscope measurement update using
   model (12b).
     $x_t \sim p(x_t|y_{1:t})$
4. If there is a new image from the camera,
   (a) Predict feature positions from the scene model using
       $\hat{x}_t = E(x_t|y_{1:t})$.
   (b) Detect the features in the image.
   (c) Measurement update with the found point correspondences
       using model (12c).
         $x_t \sim p(x_t|y_{1:t})$
5. Set $t := t + 1$ and iterate from step 2.
```

orientation estimates. This information can be used to constrain the search space when initializing from vision.

The high frequency inertial measurement updates in Algorithm 1 provide a rather accurate state estimate when a new image is acquired. This implies that the feature positions can be predicted with an improved accuracy, which in turn makes it possible to use a guided search in the image using reduced search regions. The algorithm can calculate the expected covariance of a measurement. This can be the basis for a temporal outlier removal as a complement to the spatial outlier removal provided by RANSAC methods [12]. Alternatively it can be used to predict the quantity of new information that a certain feature can contribute, which might be useful for task scheduling when the computational resources are limited [9].

The camera pose is estimated implicitly by Algorithm 1 rather than trying to determine it explicitly by inverting the measurement equations. Hence, when sufficient motion is present, the system is able to continue tracking with a very low number of features and maintain full observability using temporal triangulation.

The information from the IMU makes Algorithm 1 robust for temporary absence of vision. Without vision measurements the estimates will eventually drift away. However, short periods without vision, for instance, due to motion blur, obstruction of the camera or an unmodeled scene, can be handled without problems.

Finally, Algorithm 1 is rather flexible. It can be rather straightforwardly extended to include other information sources. For instance, a GPS might be added to aid with outdoor applications.

4 Implementation considerations

When implementing Algorithm 1, several practical issues have to be solved. These turn out to be critical for a successful system, motivating their treatment in this section.

4.1 Metric scale

As mentioned in Sect. 2.3, vision-only methods suffer from a scale ambiguity, since projections, unit-less measurements, are used. Once the scale of the scene model is defined, camera pose can be determined explicitly using three or more correspondences in combination with a calibrated camera. However, changing the scale of a scene model will give scaled, but indistinguishable poses. Hence, for vision-only applications scene models can have an arbitrary scale; a standard choice is to define the unit length to be the distance between the first two cameras.

For the inertial–vision combination, the scale is relevant. Sensor fusion utilizes position information both from the camera and the IMU, which implies that these quantities must have identical units. Scale is also important when assumptions are made about the motions of the camera, for instance the type and parameters of a motion model [10].

Introducing a metric scale into the scene model solves this issue. An existing scene model with arbitrary scale can be converted by comparing it with a *Computer Aided Design* (CAD) model or measuring an object with known dimension. An interesting solution might be to include metric information, for instance using accelerometers, in the algorithms for building the scene models. However, this is still an open question.

4.2 Vertical alignment

Accelerometers cannot distinguish accelerations of the body from gravity, as previously discussed in Sect. 2.2. To separate the contributions in the measurement, the gravity vector can be rotated from the earth coordinate system to the body frame and then subtracted. Hence, the scene model should be vertically aligned, or equivalently the gravity vector should be known in the scene model. Typically, this is not the case.

The performance of the system is extremely sensitive to this alignment, since gravity is typically an order of magnitude larger than normal body accelerations. For example, a misalignment of 1° introduces an artificial acceleration of 0.17 m/s^2 which gives rise to a systematic position drift of 8.5 cm when integrated over 1 s. Hence, even for small errors a systematic drift is introduced which causes the system to lose track without continuous corrections from correspondences. In this case the drift followed by a correction gives rise to a sawtooth pattern in the estimates, which deteriorates performance and will be visible as 'jitter'.

The gravity vector can be determined by averaging the accelerometer readings over some time, while the camera is stationary in a known pose. However, a preferable method is to record accelerometer measurements while scanning the scene and include this data in the model building procedure to align the scene model vertically.

4.3 Sensor pose calibration

The camera and the IMU both deliver measurements which are resolved in the camera and the body coordinate system, respectively. Typically, these do not coincide, since the sensors are physically translated and rotated with respect to each other. This rigid transformation should be taken into account while fusing the measurements.

The problem of determining the relative position and orientation is a well studied problem in robotics where it is known as hand–eye calibration, see e.g., Strobl and Hirzinger [38] for an introduction to this topic. However, most methods do not apply directly since the IMU does not provide an absolute position reference. Absolute orientation information is available since the accelerometers measure only gravity when the sensor unit is stationary.

The orientation part of the calibration is determined using a slight modification of standard camera calibration procedures [42], where the calibration pattern is placed on a horizontal surface and accelerometer readings are taken in the various camera poses. The camera poses are determined in the camera calibration procedure, from which the vertical directions in the camera frame can be determined. The combination of these and the vertical directions in the body frame measured by the accelerometers allows for calculation of the rotation between the frames [17, 27]. This method requires accurate positioning of the calibration pattern. As floors and desks in buildings are in practice better horizontally aligned than the walls are vertically aligned, it is recommended to use horizontal surfaces.

The translational part of the calibration is harder to estimate, and a solid calibration method which does not require special hardware is an open issue. The translation should also be available from technical drawings of the sensor unit and a rough guess using a ruler gives a quite decent result in practice. However, with increasing angular velocity this parameter becomes more dominant and an accurate calibration is necessary.

4.4 Time synchronization

It is very important to know exactly when the different measurements are taken. Multiple sensors usually have multiple clocks and these have to be synchronized. This can be achieved for instance by starting them simultaneously. However, clocks tend to diverge after a while, which will introduce problems during long-term operation. Hardware synchronization, i.e., one central clock is used to trigger the other sensors, solves this problem and this procedure has been applied in the sensor unit described in Sect. 2.

4.5 Filter tuning

The process and measurement models described in Sect. 3 have a number of stochastic components which are used to tune the filter. The settings used in the present setup are given in Table 1. The measurement noise typically depends on the sensors and should be experimentally determined. For the accelerometers and gyroscopes a measurement of a few seconds with a static pose was recorded to calculate an accurate noise covariance. Alternatively, the specification by the manufacturer can be used.

Table 1 Specifications for the sensor unit and the parameter values used for in the filter tuning

IMU	
Gyroscope range	±20.9 rad/s
Gyroscope bandwidth	40 Hz
Accelerometer range	±17 m/s^2
Accelerometer bandwidth	30 Hz
Sample rate	100 Hz
Camera	
Selected resolution	320 × 240 pixel
Pixel size	7.4 × 7.4 μm/pixel
Focal length	3.2 mm
Sample rate	12.5 Hz
Filter settings	
Gyroscope measurement noise	0.01 rad/s
Accelerometer measurement noise	0.13 m/s^2
2D feature measurement noise	0.1 pixel
3D feature measurement noise	1 mm
Angular velocity process noise	0.03 rad/s
Acceleration process noise	0.1 m/s^2
Gyroscope bias process noise	0.5 mrad/s
Accelerometer bias process noise	0.5 mm/s^2

Note that the noise parameters specify the standard deviation

The noise acting on the vision measurements is harder to determine. The algorithms return a point estimate for the obtained matches, but typically there is no stochastic information available. The accuracy for each match is highly individual and can vary a lot depending on e.g., lighting conditions, local texture, viewing angle, distance and motion blur. These individual characteristics cannot be captured by a common noise setting. Hence, it would be beneficial to include accuracy estimation in the image processing algorithms. Although attempts are being made to solve this open issue, see e.g., Skoglund and Felsberg [35], the current implementation uses a predefined noise covariance.

The process model currently used is a random walk in acceleration and angular velocity. This model is not so informative but is very general and is useful for tracking uncontrolled motions such as those generated by a human. The motion model is to be considered as a separate source of information, apart from the sensors. Hence, when more information is available in a certain application, for instance in the form of control signals, these should be included in the model to improve the filter performance. The covariances in the process model can be seen as tuning knobs, controlling the relative importance of the measurements and the process model and as such they are important parameters for stable tracking.

Valid models and parameters are imperative to obtain good estimates. The innovations, defined as the difference between a measurement and its expected value,

$$e_t = y_t - \hat{y}_t, \tag{13}$$

can be used to asses whether the models are correctly tuned. Under the model assumptions, the innovations should be normally distributed and the squared normalized innovations $e_t^T S_t^{-1} e_t$, where S_t is the predicted covariance of the measurement, should have a χ^2 distribution. It is highly recommendable to monitor these performance indicators, especially during testing, but also during normal operation.

5 Experiments

This section is concerned with an experiment where Algorithm 1 with an EKF is used to fuse the measurements from the sensor unit in order to compute estimates of its position and orientation. The experimental setup is discussed in Sect. 5.1 and the performance of the proposed inertial–vision combination provided by the sensor unit is assessed in Sect. 5.2.

5.1 Setup

The sensor unit is mounted onto a high precision 6 degrees of freedom (DoF) ABB IRB1440 industrial robot, see Fig. 3. The reason for this is that the robot will allow us to make repeatable 6 DoF motions and it will provide the true position and orientation. The robot has an absolute accuracy of 2 mm and a repeatability of 0.2 mm. This enables systematic and rather objective performance evaluation of various algorithms, based on absolute pose errors instead of the commonly used feature reprojection errors. The sensor unit provides 100 Hz inertial measurements synchronized with 12.5 Hz images. The complete specification is listed in Table 1. The scene used for the experiments consists of two orthogonal planar surfaces as shown in Fig. 2. Because of the simple geometry, the scene model could be constructed from a textured CAD model. Its coordinate system is such that the x-axis points upward and that the y- and z-axes span the horizontal plane. Although the scene was carefully positioned, it had to be calibrated w.r.t. gravity as described in Sect. 4.2. It should be emphasized that the scene has been kept simple for experimentation purposes only. The system itself can handle very general scenes and these are modeled using the methods described in Bartczak et al. [24].

With the setup several trajectories have been tested. In this paper, an eight-shaped trajectory, shown in Fig. 4, will be discussed in detail. The sensor unit traverses this 2.6 m eight-shaped trajectory in 5.4 s, keeping the scene in view at all times. The motion contains accelerations up to 4 m/s^2 and angular velocities up to 1 rad/s. Hence, the motion is quite aggressive and all 6 DoF are exited. As the displacement between images is limited to 15 pixels it is still possible to use vision-only tracking, which allows for a comparison between tracking with and without an IMU. The experiment starts with a synchronization motion, which is used to synchronize the ground truth data from the industrial robot with the estimates from the system. Time

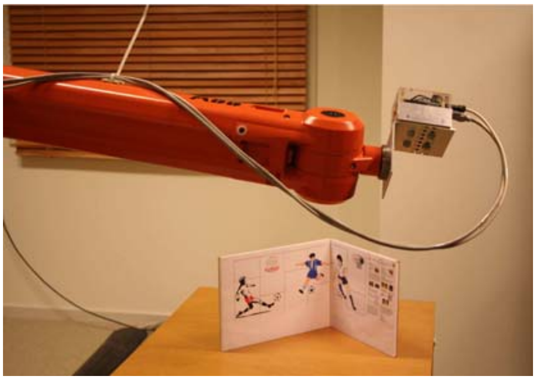

Fig. 3 The camera and the IMU are mounted onto an industrial robot. The *background* shows the scene that has been used in the experiments

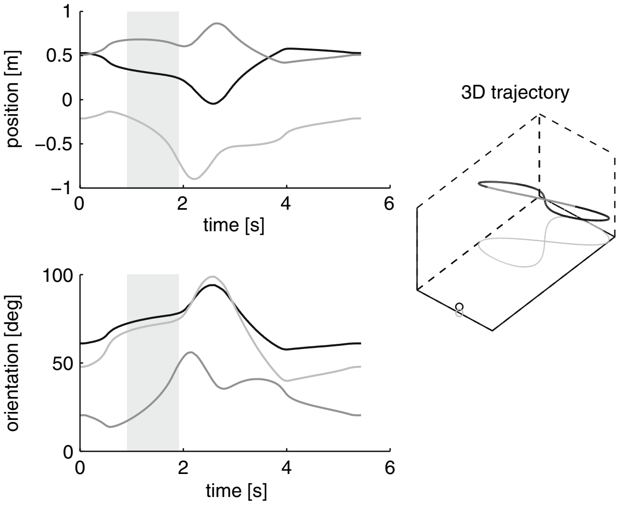

Fig. 4 The eight-shaped trajectory undertaken by the sensor unit. The *gray shaded parts* mark the interval where vision is deactivated. The *circle* indicates the origin of the scene model

synchronization is relevant, since a small time offset between the signals will result in a significant error. After the synchronization, the eight-shaped trajectory (see Fig. 4) is repeated several times, utilizing the accurate and repeatable motion provided by the industrial robot.

5.2 Results

The experimental setup described in the previous section is used to study several aspects of the combination of vision and inertial sensors. The quality of the camera pose estimates is investigated by comparing them to the ground truth data. Furthermore, the increased robustness of the system is illustrated by disabling the camera for 1 s during the second pass of the eight-shaped trajectory. Additionally, the feature predictions are shown to benefit from the inertial measurements. The findings will be discussed in the following paragraphs.

By comparing the estimates from the filter to the ground truth the tracking errors are determined. Examples of position and orientation errors (z, roll) are shown in Fig. 5.

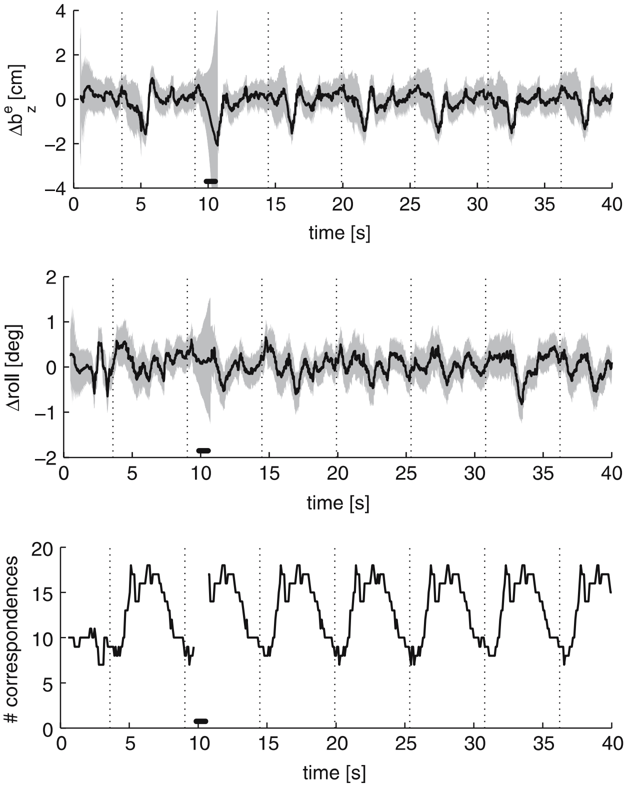

Fig. 5 Tracking error during multiple passes of the eight-shaped trajectory. The *black line* shows the position (z) and orientation (roll) errors, as well as the number of correspondences that were used. The *gray band* illustrates the 99% confidence intervals. Note that vision is deactivated from 9.7 to 10.7 s. The *vertical dotted lines* mark the repetition of the motion

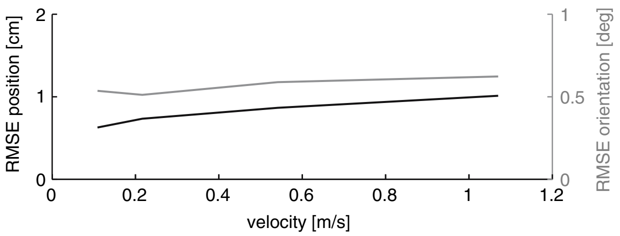

Fig. 6 Tracking error for several experiments executing the eight-shaped trajectory at different speeds

The other positions (x, y) and orientations (yaw, pitch) exhibit similar behavior. The absolute accuracy (with vision available) is below 2 cm for position and below 1° for orientation. These values turn out to be typical for the performance of the system in the setup described above. Furthermore, the accuracy of the IMU is not affected by the speed of motion, resulting in a tracking accuracy which is rather independent of velocity, as illustrated by Fig. 6 which shows the tracking error of the eight-shaped trajectory executed at various speeds. Other experiments, not described here, show similar performance for various trajectories.

A proper treatment of the implementation considerations as discussed in Sect. 4 is necessary in order to obtain good performance. Still, calibration errors and slight misalignments as well as scene model errors and other unmodeled effects are causes for non-white noise, which can deteriorate the performance. However, with the assumptions and models used, the system is shown to estimate the camera pose quite accurately using rather low-rate vision measurements. The estimated camera poses result in good and stable augmentation.

The system tracks the camera during the entire experiment, including the period where vision is deactivated. The motion during this period, indicated using gray segments in Fig. 4, is actually quite significant. Vision-only tracking has no chance of dealing with such a gap and loses track. Indeed, such an extensive period where vision is deactivated is a little artificial. However, vision might be unavailable or corrupted, due to fast rotations, high velocity, motion blur, or simply too few visible features. These difficult, but commonly occurring, situations can be dealt with by using an IMU as well, clearly illustrating the benefits of having an IMU in the system. In this way, robust real-time tracking in realistic environments is made possible.

The measurements from the IMU will also result in better predictions of the feature positions in the acquired image. This effect is clearly illustrated in Fig. 7, which provides a histogram of the feature prediction errors. The figure shows that the feature prediction errors are smaller and more concentrated in case the IMU measurement updates are used. This improvement is most significant

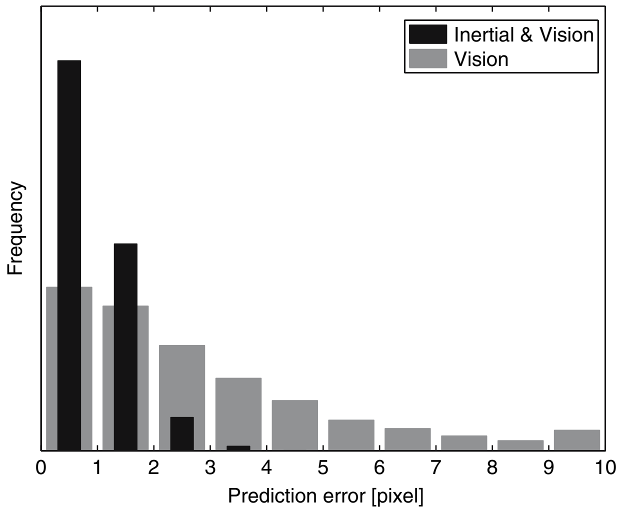

Fig. 7 Histogram of the prediction errors for the feature positions. The feature predictions are calculated using the latest vision pose and the most recent inertial pose, respectively

when the camera is moving fast or at lower frame rates. At lower speeds, the vision-based feature predictions will improve and the histograms will become more similar.

The improved feature predictions facilitate the use of smaller search regions to find the features. This implies that using an IMU more features can be detected, given a certain processing power. On the other hand, the improved feature predictions indicate that the IMU handles the fast motion and that the absolution pose information which vision provides is required at a reduced rate.

6 Conclusion

Based on a framework for nonlinear state estimation, a system has been developed to obtain real-time camera pose estimates by fusing 100 Hz inertial measurements and 12.5 Hz vision measurements using an EKF. Experiments where an industrial robot is used to move the sensor unit show that this setup is able to track the camera pose with an absolute accuracy of 2 cm and 1°. The addition of an IMU yields a robust system which can handle periods with uninformative or no vision data and it reduces the need for high frequency vision updates.

Acknowledgments This work has been performed within the MATRIS consortium, which is a sixth framework research program within the European Union (EU), contract number: IST-002013.

References

1. Armesto, L., Tornero, J., Vincze, M.: Fast ego-motion estimation with multi-rate fusion of inertial and vision. Int. J. Robot. Res. **26**(6), 577–589 (2007). doi:10.1177/0278364907079283
2. Aron, M., Simon, G., Berger, M.O.: Use of inertial sensors to support video tracking. Comput. Animat. Virtual. Worlds. **18**(1), 57–68 (2007)
3. Bergman, N.: Recursive Bayesian estimation: Navigation and tracking applications. Dissertations no 579, Linköping Studies in Science and Technology, SE-581 83 Linköping, Sweden (1999)
4. Bucy, R.S., Senne, K.D.: Digital synthesis on nonlinear filters. Automatica **7**, 287–298 (1971)
5. Chatfield, A.: Fundamentals of High Accuracy Inertial Navigation, vol. 174, 3rd edn. American Institute of Aeronautics and Astronautics, USA (1997)
6. Chroust, S.G., Vincze, M.: Fusion of vision and inertial data for motion and structure estimation. J. Robot. Syst. **21**(2), 73–83 (2004)
7. Corke, P., Lobo, J., Dias, J.: An introduction to inertial and visual sensing. Int. J. Robot. Res. **26**(6), 519–535 (2007). doi: 10.1177/0278364907079279
8. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera. In: Proceedings of 9th IEEE International Conference on Computer Vision, Nice, France, vol. 2, pp. 1403–1410 (2003)
9. Davison, A.J.: Active search for real-time vision. In: Proceedings of 10th IEEE International Conference on Computer Vision, Beijing, China, pp. 66–73 (2005)
10. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007). doi:10.1109/TPAMI.2007.1049
11. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping (SLAM): Part I. IEEE Robot. Autom. Mag. **13**(2), 99–110 (2006)
12. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6):381–395 (1981). doi: 10.1145/358669.358692
13. Gemeiner, P., Einramhof, P., Vincze, M.: Simultaneous motion and structure estimation by fusion of inertial and vision data. Int. J. Robot. Res. **26**(6), 591–605 (2007). doi:10.1177/0278364907080058
14. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-gaussian bayesian state estimation. IEE Proc. Radar Signal Process. **140**(2), 107–113 (1993)
15. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conference, Manchester, UK, pp. 147–151 (1988)
16. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
17. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. J. Opt. Soc. Am. A **4**(4), 629–642 (1987)
18. Isard, M., Blake, A.: Condensation—conditional density propagation for visual tracking. Int. J. Comput. Vis. 29(1):5–28 (1998). doi:10.1023/A:1008078328650
19. Jazwinski, A.H.: Stochastic processes and filtering theory. Mathematics in science and engineering, Academic, New York (1970)
20. Julier, S.J., Uhlmann, J.K.: Unscented filtering and nonlinear estimation, Proc. IEEE **92**(3), 401–422 (2004). doi:10.1109/JPROC.2003.823141
21. Kalman, R.E.: A new approach to linear filtering and prediction problems. Trans. ASME J Basic Eng. **82**, 35–45 (1960)
22. Kitagawa, G.: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J. Comput. Graph. Stat. **5**(1), 1–25 (1996)

23. Klein, G.S.W., Drummond, T.W.: Tightly integrated sensor fusion for robust visual tracking. Image Vis. Comput. **22**(10), 769–776 (2004)
24. Bartczak, B., Koeser, K., Woelk, F., Koch, R.: Extraction of 3D freeform surfaces as visual landmarks for real-time tracking. J. Real-Time Image Process. (this issue) doi:10.1007/s11554-007-0042-0
25. Kuipers, J.B.: Quaternions and Rotation Sequences. Princeton University Press, Princeton (1999)
26. Lobo, J., Dias, J.: Inertial sensed ego-motion for 3D vision. J. Robot. Syst. **21**(1), 3–12 (2004)
27. Lobo, J., Dias, J.: Relative pose calibration between visual and inertial sensors. Int. J. Robot. Res. **26**(6), 561–575 (2007). doi:10.1177/0278364907079276
28. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: An invitation to 3-D vision—from images to geometric models. Interdisciplinary Applied Mathematics, Springer, Berlin (2006)
29. Pieper, R.J.B.: Comparing estimation algorithms for camera position and orientation. Master's thesis, Department of Electrical Engineering, Linköping University, Sweden (2007)
30. Pinies, P., Lupton, T., Sukkarieh, S., Tardos, J.D.: Inertial aiding of inverse depth SLAM using a monocular camera. In: Proceedings of IEEE International Conference on Robotics and Automation, Roma, Italy, pp. 2797–2802 (2007). doi:10.1109/ROBOT.2007.363895
31. Ribo, M., Brandner, M., Pinz, A.: A flexible software architecture for hybrid tracking, J. Robot. Syst. **21**(2), 53–62 (2004)
32. Schmidt, S.F.: Application of state-space methods to navigation problems. Adv. Control Syst. **3**, 293–340 (1966)
33. Schön, T.B.: Estimation of nonlinear dynamic systems—theory and applications. Dissertations no 998, Linköping Studies in Science and Technology, Department of Electrical Engineering, Linköping University, Sweden (2006)
34. Shuster, M.D.: A survey of attitude representations. J. Astronaut. Sci. **41**(4):439–517 (1993)
35. Skoglund, J., Felsberg, M.: Covariance estimation for sad block matching. In: Proc. 15th Scandinavian Conference on Image Analysis (2007)
36. Smith, G.L., Schmidt, S.F., McGee, L.A.: Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. Tech. Rep. TR R-135, NASA (1962)
37. Stricker, D., Thomas, G.: The MATRIS project: real-time markerless camera tracking for AR and broadcast applications. J. Real-Time Image Processing. doi:10.1007/s11554-007-0041-1
38. Strobl, K.H., Hirzinger, G.: Optimal hand-eye calibration. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp. 4647–4653 (2006). doi:10.1109/IROS.2006.282250
39. Thrun S., Burgard W., Fox D. (2005) Probabilistic Robotics. Intelligent Robotics and Autonomous Agents. The MIT Press, Cambridge
40. Titterton, D.H., Weston, J.L.: Strapdown inertial navigation technology. IEE radar, sonar, navigation and avionics series. Peter Peregrinus, Stevenage (1997)
41. Williams, B., Smith, P., Reid, I.: Automatic relocalisation for a single-camera simultaneous localisation and mapping system. In: Proceedings of IEEE International Conference on Robotics and Automation, Roma, Italy, pp. 2784–2790 (2007). doi:10.1109/ROBOT.2007.363893
42. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000). doi:10.1109/34.888718

Author Biographies

Jeroen D. Hol was born in the Netherlands in 1981. He recieved the M.Sc. in Mechanical Engineering at the University of Twente, Enschede, the Netherlands. Currently he is a Ph.D. student at the Division of Automatic Control at the Department of Electrical Engineering at Linköping University, Linköping, Sweden. His research is focused on sensor fusion and signal processing, with applications in Augmented Reality.

Thomas B. Schön was born in Sweden in 1977. He received the B.Sc. degree in Business Administration and Economics in Feb. 2001, the M.Sc. degree in Applied Physics and Electrical Engineering in Feb. 2001 and the Ph.D. degree in Automatic Control in Feb. 2006, all from Linköping University, Linköping, Sweden. He has held visiting positions at the University of Cambridge (UK) and the University of Newcastle (Australia). His research interests are mainly within the areas of signal processing, sensor fusion and system identification, with applications to the automotive and aerospace industry. He is currently a Research Associate at Linköping University.

Henk Luinge received the M.Sc. degree in mechanical engineering in 1998 and the Ph.D degree in Electrical engineering in 2002 both from Twente University, Enschede, the Netherlands. He was a postdoctoral fellow at the Rehabilitation institute of Chicago as well as the Ècole Nationale d'Alfort, Paris, France. He is currently leading the Xsens research team, developing algorithms for systems containing miniature inertial sensors in areas such as biomechanics, robotics and navigation.

Per J. Slycke received the M.Sc. degree in physics from the University of Twente, Enschede, the Netherlands in 1999. In 2000 he co-founded Xsens Technologies B.V., Enschede, a company specialized in inertial based 3-D motion tracking technology. As CTO of Xsens his research interests include novel developments in (MEMS) inertial sensors, stochastic signal processing for inertial sensors in combination with various aiding sensor technologies such as GNSS, magnetometers, computer vision, as well as applications for robotics and unmanned crafts as well as biomechanics and rehabilitation.

Fredrik Gustafsson received the M.Sc. degree in Electrical Engineering in 1988 and the Ph.D. degree in Automatic Control in 1992, both from Linköping University, Linköping, Sweden. He is professor of sensor informatics with the department of Electrical Engineering, Linköping University. His research is focused on sensor fusion and statistical methods in signal processing, with applications to aerospace, automotive, audio and communication systems. He is the author of four books, over hundred international papers and fourteen patents. He is also a co-founder of three spin-off companies in these areas. Prof. Gustafsson is an associate editor of IEEE Transactions on Signal Processing.