A Flight State Estimator that Couples Stereo Vision, INS, and GNSS Pseudo-Ranges to Navigate with Three or Less Satellites

Franz Andert, Jörg Dittrich, Simon Batzdorfer, Martin Becker, Ulf Bestmann, Peter Hecker

Abstract This paper presents a flight state estimator which couples stereo vision, inertial (INS), and global navigation satellite system (GNSS) data. The navigation filter comes with different operation modes that allow loosely coupled GNSS/INS positioning and, for difficult conditions, improvements using visual odometry and a tighter coupling with GNSS pseudo-range (PSR) data. While camera systems are typically used as an additional relative movement sensor to enable positioning without GNSS for a certain amount of time, the PSR data filtering allows to use satellite navigation also when less than four satellites are available. This makes the filter even more robust against temporary dropouts of the full GNSS solution. The application is the navigation of unmanned aircraft in disaster scenarios which includes flights close to ground in urban or mountainous areas. The filter performance is evaluated with sensor data from unmanned helicopter flight tests where different conditions of the GNSS signal reception are simulated. It is shown that the use of PSR data improves the positioning significantly compared to the dropout when the signals of less than four satellites are available.

1 Introduction

Positioning and navigation with limited satellite reception is one of the current challenges for unmanned vehicles. Global satellite navigation has its known drawbacks such as a varying accuracy due to the satellite constellation, atmospheric errors, or possible signal interruption and reflection. Unmanned aircraft navigation becomes

Franz Andert, Jörg Dittrich
German Aerospace Center (DLR), Institute of Flight Systems, 38108 Braunschweig, Germany
e-mail: franz.andert@dlr.de, joerg.dittrich@dlr.de

Simon Batzdorfer, Martin Becker, Ulf Bestmann, Peter Hecker
Technische Universität Braunschweig, Institute of Flight Guidance, 38108 Braunschweig, Germany
e-mail: s.batzdorfer@tu-bs.de, m.becker@tu-bs.de, u.bestmann@tu-bs.de, p.hecker@tu-bs.de

problematic especially in the proximity of ground objects, for example in flights through urban or natural canyons. Especially such scenarios require abilities to reduce the positioning uncertainty for safe flights without collisions. The combination of satellite navigation (GNSS), such as GPS or the upcoming Galileo system, with inertial systems (INS) is quite common. But the ability to compensate longer satellite signal dropouts depends on the accuracy and drift rates of the INS, and the available technology for small and lightweight unmanned aircraft is presently insufficient [6].

Fig. 1 DLR's 13 kg helicopter with a stereo camera, onboard image processing and GNSS/INS navigation filtering.

The application context of this paper is low-altitude outdoor exploration flights in disaster scenarios with the unmanned helicopter shown in fig. 1. Since cameras are often on board these vehicles for various applications and their motion can be obtained from image sequences, it is straightforward to use them for improving the navigation solution here as well. The developed solution should be able to be run under difficult conditions and also in unknown areas, this is why the usage of a-priori knowledge from maps as proposed in [11, 14] is not suitable here. With that, only relative movements are determinable from the camera images so that the presented solution will be influenced by accumulating errors as soon as satellite navigation becomes unavailable. Contrary to many other approaches, this paper does not address scaling issues that come with monocular cameras being solved by integrating inertial measurements into the motion estimator [22, 27] which determines the scale with respect to the observed accelerations, or by using additional sensors like a barometer [1] or distance sensors like laser scanners as proposed by [25]. Instead, this approach uses a calibrated stereo camera to determine relative 3D movements and rotations. Similar approaches are used in [10, 13, 16]. However, the developed filter might be further improved by the mentioned related work so that laser scanners, monocular cameras, barometers, and many other sensors can be combined with the presented filter instead of the stereo camera.

Beside the usage of a camera as an additional sensor for the GNSS/INS navigation filter, the paper addresses the problem of partial GNSS signal dropouts. This concerns signal receptions of three or less satellites which do not give the complete position information by themselves. In these cases, a classical loosely coupled GNSS position support of drifting inertial data would fail. However, the PSR data give some hints about the position (e.g. a line in the case of three satellites) which can be matched with the information from the inertial and vision system [29]. If now the position can be recovered, the proposed navigation filter reduces the chance of positioning dropouts, especially in cluttered outdoor environments where the number of visible satellites may be often low.

2 Related Work

The idea of including visual information into a navigation filter to localize a vehicle in obstacle-prone or indoor environments is very promising. Within the last years, many technical advances have been evolved from the off-the-shelf availability of small and easily manageable aircraft (like quadrotors) and lightweight cameras and computers with sufficient performance. This section gives a brief overview of the different ideas that act as a basis for the principles developed in this paper.

2.1 Image Processing and Visual Odometry

Camera motion estimation is generally based on the motion visible in the image sequences. This requires a scenery with mostly unmoved objects within the camera's field of view and some identifiable patterns to find homologous points in the images representing the same stationary object points. Technically, this refers to identifying a sparse set of homologous points determinable by feature detection and tracking algorithms. In the stereo image case, corresponding points between two image pairs are to be identified with the advantage that the absolute scale of motion is determinable if the objects are within the usable range of stereo-based distance measurements.

One common visual odometry principle breaks the camera positioning down to a camera motion or relative orientation estimation between two images or image pairs. For the stereo case that produces 3D image features, the transformation between the two resulting point clouds can be determined by general registration algorithms [5, 8] or those optimized for stereo vision [17]. The results are camera pose updates that can be integrated into the camera trajectory. The easiest way would now be to incrementally integrate all succeeding images, but due to the large amount of little erroneous steps, the accumulation error will be rather large. Better results are to be expected when keyframes are used. For each update step, this means to look back

in the image and feature history and find the oldest point cloud with enough overlap with the newest one so that a registration is still possible.

The other very common principle has its roots in photogrammetric triangulation and resectioning as well as in simultaneous localization and mapping (SLAM), see [25] for a recent overview within the aerial robotics domain. The idea is to project the observed image points of all images into the same 3-D coordinate system, and to get the current camera position by the registration between this map and the current image points. The correspondences between map and image points are usually known since all map points have been derived from the previous images. In addition to this registration, the current image points are fused to the map, which means to add new points and to reduce the statistical errors of the existing ones.

2.2 Vision-Based Navigation Filtering

Based on the different methods to measure the camera's motion, there are a lot of ways how vision data are integrated into INS or GNSS/INS navigation filters. All presented approaches have in common that they are based on a constant and known (but potentially biased) camera alignment on the vehicle. With that, camera measurements can be transformed into the vehicle-relative system, yielding a sensor that serves vehicle motion components.

An early approach for UAVs that is completely integrated into the navigation and control system is presented in [18]. A monocular, downward-looking camera is used, and image feature positions are directly integrated as measurements into an Extended Kalman Filter (EKF) which estimates the flight state. This is simple since no registration or other camera orientation estimation algorithms are applied, but also effective since it is shown that GPS dropouts of more than one minute can be handled including the stable control of an unmanned helicopter. However, the approach only integrates in-plane translational movements from the camera images and assumes a correct measurement of the ground distance.

The more complex image processing procedures such as the mentioned relative motion estimation and SLAM methods are also promising to enhance existing flight estimators. The main difference between both techniques is that relative motion consists of translational and angular movements between two images and acts more as a kind of speed sensor to be integrated into the flight state estimator [27]. Contrary to that, SLAM directly returns the camera pose based on the current and all previous images, and especially the position seems to be integrable into the navigation filter [1]. Both methods have been successfully integrated into aerial systems. However, all of these methods, including the direct use of image feature positions, are all affected by accumulating errors over time. An exception is the case where longer back-dated features are still visible in the current image so that a direct registration is possible. This means for the practical use that accumulating errors can theoretically be eliminated as long as the vehicle is in hovering mode. For example, this is confirmed by [3] where a vision-based and drift-free hover stabilization is presented.

2.3 Positioning with a Low Number of Satellites

Navigation with three satellites would be easy if a precise, i.e. atomic clock is present [23]. Since this is not an option for small unmanned aircraft and due to the cost not even for most other applications, the data from other sensors are to be brought in to get a plausible estimation of the current position. GNSS/INS navigation filters can generally handle dropouts of the GNSS signal by integrating inertial data. For example, it is shown in [28] that the drift can be significantly reduced when GNSS/INS raw data such as pseudo-ranges and phase information are tightly coupled into the state estimator. Nevertheless, this is not free from drift given incomplete satellite reception, which might still be insufficient for UAV navigation where no tactical grade inertial measurement units are used.

3 Image-Based Motion Estimation

This section describes the method to determine relative motion estimates from camera image sequences. As the principles are widely known from the related literature, the basics are only briefly introduced. Focus of this section are supplementary implementations, explained in more detail.

3.1 Determining Optical Movements

Visible disparities of a set of selected image points $\{\mathbf{p} : (x,y)^\top\}$ are the basis for camera movement determination, see fig. 2. Based on an initial point set $\{\mathbf{p}_t^L\}$ for the left camera from corner detection or the previous tracking step, a tracking over time $\{\mathbf{p}_t^L\} \rightarrow \{\mathbf{p}_{t+1}^L\}$ is performed, here with the *Lucas-Kanade* algorithm [19]. Stereo comparison for every time stamp $\{\mathbf{p}_t^L\} \rightarrow \{\mathbf{p}_t^R\}$ is done in an analogous way. Following stereoscopic math (e.g. [20]), this returns a point cloud with 3D features $\{\mathbf{q} : (x,y,z)^\top\}$ in camera coordinates for every image frame time stamp t. For further calculations, these point clouds are transformed to the vehicle coordinate frame by using an initially measured camera alignment.

3.2 Ego-Motion Estimation

Generally, the relative motion is denoted as a 4×4 transformation matrix $\mathbf{T}$ containing the 3×3 rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ in the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{t}_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}. \tag{1}$$

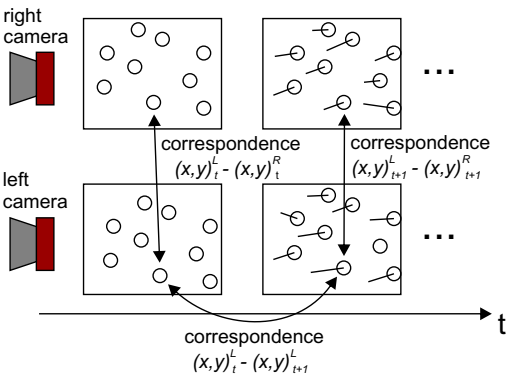

Fig. 2 Relationship between the image sequence and the optical movements. The correspondence of points is determined between different images of the left camera (temporal movement) and between the two images from the cameras (spatial disparities). The circles and the lines show the feature points and their movement vector.

Let the matrix $\mathbf{T}_{t_1:t_2}$ define the relative movements between the time stamps t_1 and t_2. From stereo image points, it is determined through the rigid transform of the point clouds $\{\mathbf{q}_{t_1}\}$ and $\{\mathbf{q}_{t_2}\}$. Several algorithms which do or do not require initial correspondences have been tested to do this job, and it turned out that the iterative closest point (ICP) algorithm with nonlinear optimization backend from the *Point Cloud Library* [24] performs best if no other hints like inertial data are given. Since the correspondence of points $[\mathbf{q}_{t_1}, \mathbf{q}_{t_2}]$ is known from feature tracking, the input point clouds are reduced to the corresponding elements that exist in both. This avoids false convergence and improves the estimation results.

The fitness of the transformation is returned by the error covariance matrix $\mathbf{Cov}(\{\mathbf{T}_{t_1:t_2}\mathbf{q}_{t_1}\},\{\mathbf{q}_{t_2}\})$ based on point distances between the point clouds transformed to the same coordinate system. Here, the sets $\{\mathbf{q}_{t_1}\}$ and $\{\mathbf{q}_{t_2}\}$ only include the points that remain relevant for transformation estimation, i.e. outliers are not included. The matrix elements are e.g.

$$\mathrm{cov}_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_{t_1,i} - x_{t_2,i})(y_{t_1,i} - y_{t_2,i}), \tag{2}$$

the other matrix elements are calculated analogously. Since the centroids of both transformed point clouds are equal after an ICP transform, the mean distance is zero and omitted in the equation.

3.3 Using Key Frames

While classic visual odometry is based on incremental transformations $\mathbf{T}_{t-1:t}$, the usage of key frames means to estimate the transformation $\mathbf{T}_{t-\tau:t}$ between the current frame t and the oldest possible frame $t-\tau$ (i.e. τ frames older) from the image sequence. As already mentioned, this will presumably reduce the accumulating error. Transformations are determinable as long as an overlap exists between an older image and the current one. Practically, this refers to the availability of corresponding

homologous points which were already detected in both images. Fortunately, this does not mean that the image sequence has to be saved. It is sufficient to store the features $\{\mathbf{q}_t\}$ of the images. Because the probabilities of a successful transformation determination between the current image and two older ones with the same feature set are supposedly similar, it is further sufficient to store the features of those images where new features are added and mark them as a key frame. This results in a reduced set of key frames with only the oldest frame from each sub-sequence where the images have all the same tracked features.

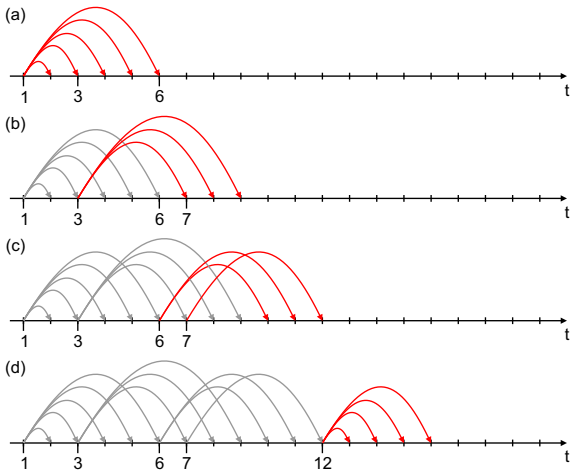

Fig. 3 Using key frames to determine camera movements. Comparing every new frame with the first one (a), jumping to the next key frames (b,c), using the last frame as key if no other key frame is available (d). The curved arrows show which frames are being compared during the sequence.

Figure 3 illustrates how the key framing process works. In (a), the ideal case is shown where the current image features are compared to the features from image 1. Every time new features are added to the tracking list, a key frame (i.e. the list of features) is stored. In the shown example, key frames from the time stamps 3 and 6 are saved. This does not mean that they are immediately used for comparison with the next frames. Older key frames will still be used as long as the transformation calculation is successful.

A key frame is not valid anymore and will be deleted if the number of feature points having corresponding ones in the current frame becomes too low, or if the attempt to estimate the relative transformation fails. In these cases, the next oldest possible key frame will be used until a valid transformation returns. An example is shown in (b) where frame 3 is used from now on as the oldest valid key frame. The advantage of storing multiple key frames is shown in (c) where frame 6 is used as the key frame for the time step 10 and 11, and after it becomes invalid, another rather old frame 7 is available at time step 12. Beyond that, subfigure (d) shows the case where no valid key frame is available at time step 13 and the last frame 12 will be used as the new key from now on. Nevertheless, it remains the rather unlikely case where no transformation to older frames is determinable at all. Resulting odometry gaps are handled by the navigation filter.

3.4 Using Feedback of the Predicted Flight State

In the presented setup illustrated in fig. 6, the predicted flight state is coupled back to estimate relative motions. With the states corresponding to the images, the point clouds are transformed to geodetic coordinates, and the ego-motion estimation directly returns the geodetic movement and rotation. In addition to that, the rotation is already obtained from inertial data (i.e. both geodetic point clouds do not have any rotation to each other in the ideal case), and it is sufficient to estimate the translational movement. This can easily be done by calculating the difference of the point cloud centroids. Combined with an outlier filtering that removes corresponding points with large distances remaining after transformation, this returns the camera movement $\mathbf{t}_{t_1:t_2}$. In the results section, it is shown from recorded image data that this performs better than the estimation of all six degrees of freedom as described before. Therefore, only the translational movement is being coupled with the flight state estimation filter.

4 State Estimation

The flight state estimation follows the common principles described in [7, 9] which is a part of the navigation research at the TU Braunschweig. The state $\mathbf{x}$ is denoted as the vector

$$\mathbf{x} = (\mathbf{p}^\top, \mathbf{v}^\top, \mathbf{q}^\top, \mathbf{b}_a^\top, \mathbf{b}_\omega^\top)^\top \tag{3}$$

with WGS84 position vector $\mathbf{p}_{(3\times1)}$ (latitude ϕ, longitude λ, ellipsoidal height h), velocity vector $\mathbf{v}_{(3\times1)}$, attitude quaternion $\mathbf{q}_{(4\times1)}$, and biases of acceleration $\mathbf{b}_{a\,(3\times1)}$ and turn rates $\mathbf{b}_{\omega\,(3\times1)}$.

The Extended Kalman Filter (EKF) loop to estimate $\mathbf{x}$ contains the high-frequency time update, i.e. the prediction step with inertial data

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \\ \mathbf{P}_k^- &= \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1}\boldsymbol{\Phi}_{k-1}^\top + \mathbf{Q} \end{aligned} \tag{4}$$

with the predicted state $\hat{\mathbf{x}}_k^-$ and its covariance $\mathbf{P}_k^-$ at step k based on the previous estimation and the input vector from inertial data $\mathbf{u}_k$ (see sec. 4.1). Lower-frequency GNSS and vision data are measurement vectors $\mathbf{z}_k$ (see sec. 4.2 and 4.3). If $\mathbf{z}_k$ is available, the correction step is

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}_k^\top(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^\top + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-)\right) \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \end{aligned} \tag{5}$$

yielding the estimator gain matrix $\mathbf{K}_k$, the corrected estimation $\hat{\mathbf{x}}_k$ and its covariance matrix $\mathbf{P}_k$. The other symbols are: measurement matrix $\mathbf{H}_k$, nonlinear measurement

function $h(\hat{\mathbf{x}}_k^-)$, transition matrix Φ_k, process noise covariance matrix $\mathbf{Q}$, and measurement noise covariance matrix $\mathbf{R}_k$. There are two different update steps based on either vision or GNSS, which are run in succession if data from both are available. This means that the vision update step yields a new state prediction, which is updated by GNSS data afterwards. Detailed explanations about the prediction and update steps are given in the next subsections.

4.1 Prediction with Strapdown Calculation

The inertial system measures three-dimensional body-fixed accelerations and turn rates. This defines the vector $\mathbf{u} = (\mathbf{a}, \omega)$. Based on this information, the earth's gravity and known initialization values for position, velocity, and attitude, the so-called strapdown calculation $f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$ returns new values for every time stamp. The calculation consists of differential equations implemented in the navigation software. Basically, it integrates the accelerations to velocities and twice to positions and the turn rates to the attitude angles. Further details such as the compensation for the earth's rotation and equations are given in [7].

4.2 Update with Image Data

As already mentioned, it turned out that visual odometry performs best when the estimated flight state is coupled back to the image processing, which will then estimate only position differences between two images. Let the vector $\mathbf{t}_{t_1:t_2}$ (see eq. 1) be the estimated motion, $\mathbf{p}_{t_1}$ "plus" $\mathbf{t}_{t_1:t_2}$ would return the current position estimate. Although the cameras are triggered by the navigation clock based on inertial and GNSS data, the times t_1 and t_2 may differ slightly from the filter update time stamps. Hence, the closest time stamps t_{k_1} and t_{k_2} of filter updates k_1 and k_2 are the basis for the measurement

$$\mathbf{z}_k = \mathrm{p2w}(\mathbf{t}_{t_1:t_2} \cdot \frac{t_{k_2} - t_{k_1}}{t_2 - t_1}, \mathbf{p}_{k_1}) \tag{6}$$

of the image-based position. The function $\mathrm{p2w}(\mathbf{t}, \mathbf{p}_0)$ denotes the function which converts a Cartesian coordinate translation vector $\mathbf{t}$ relative to a fundamental point $\mathbf{p}_0$ to the geodetic system [21]. Since the values of $\mathbf{z}_k$ directly give the position, the corresponding measurement matrix is constant

$$\mathbf{H}_k = \begin{pmatrix} 1\,0\,0\,0\,\ldots\,0 \\ 0\,1\,0\,0\,\ldots\,0 \\ 0\,0\,1\,0\,\ldots\,0 \end{pmatrix} \tag{7}$$

and accordingly, $h(\hat{\mathbf{x}}_k^-)$ returns the first three values of the vector $\hat{\mathbf{x}}_k^-$.

The matrix $\mathbf{R}_k$ is taken from the covariance matrix as described in sec. 3.2. Like the motion, it is "stretched" slightly with the quotient of the different time durations. It is

$$\mathbf{R}_k = \left(\frac{t_{k_2} - t_{k_1}}{t_2 - t_1} \right)^2 \mathbf{Cov}(\{\mathbf{T}_{t_1:t_2}\mathbf{q}_{t_1}\}, \{\mathbf{q}_{t_2}\}). \tag{8}$$

4.3 Update with GNSS Data

The update step uses GPS pseudo-range data and follows the principles presented e.g. in [12, 15]. Details of this method are described in the literature, the basic approach is the following: For the i-th satellite ($i = 1, \ldots, n$), the used data include the pseudo-range ρ_i, its standard deviation σ_i as well as the time errors and satellite positions from the ephemeris data. To include this into the state filter, the measurement vector $\mathbf{z}_k$ is built by the measured ranges, it is

$$\mathbf{z}_k = (z_1, z_2, \ldots, z_n)_k^\top \tag{9}$$

containing the corrected pseudo-ranges from the n visible satellites. It is

$$z_i = \rho_i - c_0 \left(\Delta t - t_{\text{Sat},i} + \Delta t_{\text{tropo},i} + \Delta t_{\text{iono},i} \right) \tag{10}$$

with the measured pseudo-ranges $\rho_1, \rho_2, \ldots, \rho_n$ and the time differences from clock, tropospheric, and ionospheric errors multiplied by the speed of light c_0.

The observation matrix $\mathbf{H}$ for the k-th update is the Jacobian matrix

$$\mathbf{H}_k = \begin{pmatrix} \frac{\partial \rho_1}{\partial \phi} & \frac{\partial \rho_1}{\partial \lambda} & \frac{\partial \rho_1}{\partial h} & 0 \ldots 0 \\ \vdots & \vdots & \vdots & \vdots \ddots \vdots \\ \frac{\partial \rho_n}{\partial \phi} & \frac{\partial \rho_n}{\partial \lambda} & \frac{\partial \rho_n}{\partial h} & 0 \ldots 0 \end{pmatrix} \tag{11}$$

of pseudo-range derivatives with respect to the geodetic position.

To map $\hat{\mathbf{x}}_k^-$ onto a predicted measurement, the function $h(\hat{\mathbf{x}}_k^-)$ returns a vector $\hat{\mathbf{z}}_k^- = \left(\hat{z}_1^-, \hat{z}_2^-, \ldots, \hat{z}_n^-\right)_k^\top$ with the predicted pseudo-ranges. It is the Euclidean distance

$$\hat{z}_i^- = \left\| \mathbf{p}_{\text{Sat},i} - \text{w2e}(\hat{\mathbf{p}}^-) \right\|_2 \tag{12}$$

between the cartesian earth-centered and earth-fixed (ECEF) i-th satellite position $\mathbf{p}_{\text{Sat},i}$ and the result of the function $\text{w2e}(\hat{\mathbf{p}}^-)$ which converts the predicted WGS84 position into the ECEF system [21].

The error covariance matrix $\mathbf{R}_k$ is defined by the standard deviations σ_i of the pseudo-ranges, it is

$$\mathbf{R}_k = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2)_k. \tag{13}$$

5 Experimental Setup

Goal of this work is the implementation of a vision-aided navigation filter and its evaluation during flight test. This section gives an overview about the hardware, general software architecture, and the reference system for validation.

5.1 Flight Hardware

Testing vehicle (fig. 4) is the 13 kg helicopter ARTIS (Autonomous Rotorcraft Testbed for Intelligent Systems) of the DLR Institute of Flight Systems [2]. The navigation sensors are a ublox-6 GPS as the GNSS receiver and a custom-built IMU with two 2-axis accelerometers (Bosch SMB 225) and and three 1-axis gyros (Bosch SMG 074) including calibration and temperature compensation. Image sensors are

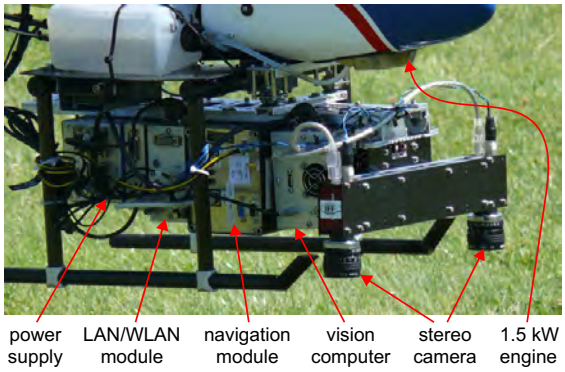

Fig. 4 ARTIS helicopter with navigation and image processing payload.

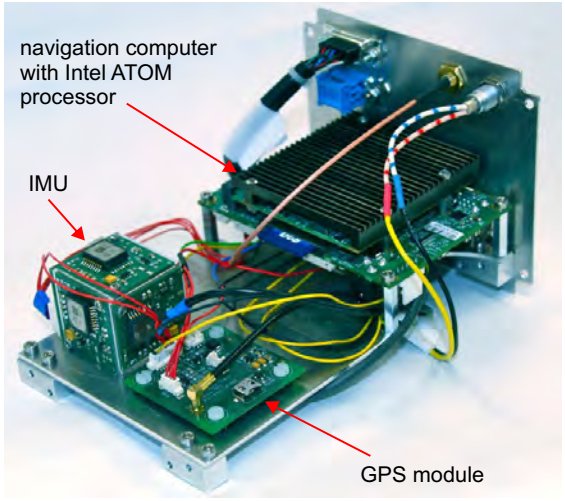

Fig. 5 The navigation module with its internal components.

two digital global shutter firewire cameras (AVT Marlin F131B, resolution: max. 1280×960 px, framerate: max. 30 Hz, lens/focal length: 1265 px) with a baseline of 30 cm. Synchronous image exposures are triggered with a signal based on the pulse per second output from the GPS receiver. The navigation module (fig. 5) combines GPS, INS, and navigation computer in a single box, separated from the image processing computer.

5.2 Processing Software Architecture

As navigation and image processing are separated by the hardware, the tightly coupled filter is based on two software frameworks with bidirectional data exchange. Fig. 6 shows the core components.

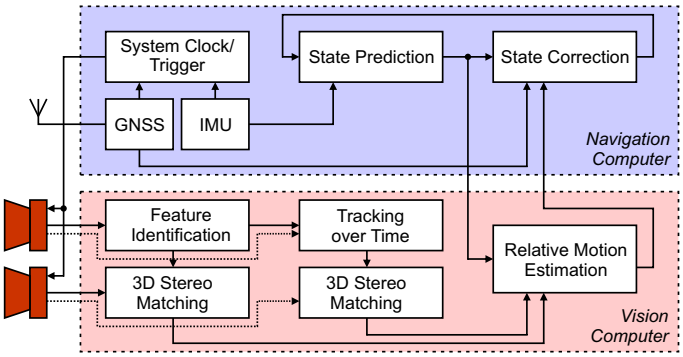

Fig. 6 Navigation and image processing software components.

The flight state estimator follows Extended Kalman Filter (EKF) principles with a state prediction based on high-frequency inertial measurements (here: 100 Hz update) and a state correction with measurements from sensors with lower update rates. These are the GPS pseudo-range (PSR) data and the camera-based relative motion. Relative motions are estimated with six degrees of freedom (DoF) onboard the image processing computer from homologous image points that are identified with a feature tracking algorithm. The relative motion estimation is additionally supported by the predicted flight state.

5.3 Reference Measuring

The computed solution with full and simulated limited GNSS reception is referenced to an augmented high-precision positioning based on the raw satellite data and post-processing corrections from state survey services [26]. The reference position has an accuracy of few centimeters when the availability of satellite data is

sufficient as in the flight tests on a model aircraft flight field. As anticipation for future flights in urban environments with real signal dropouts, laser-based tracking and measuring was also established as a reference independently from GNSS signals [4].

6 Flight Testing and Evaluation

The presented methods are tested with data recorded during remotely controlled helicopter flights over a model airfield with a grass runway, some vegetation, and a small house at one side where the ground control station car is parked. The cameras are looking downward, example images are shown in fig. 7. In the following section, the development steps are tested constructively. First, it is shown whether the usage of key frames from sec. 3.3 is a useful procedure within visual odometry. Second, the visual relative movement calculation is improved with the feedback from the flight state prediction from sec. 3.4. And finally, these movements are forwarded to the state estimator from sec. 4.

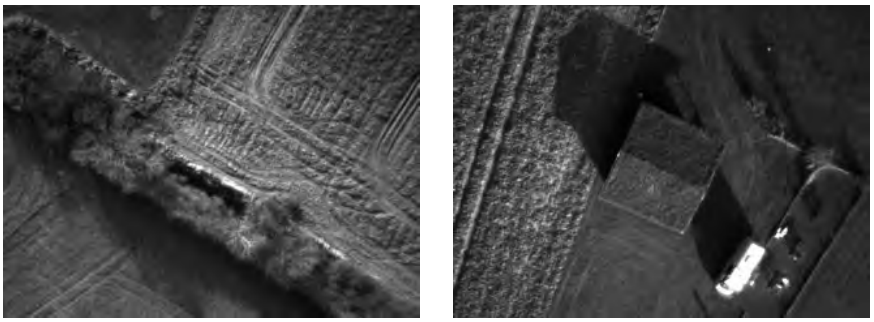

Fig. 7 Examples of the analyzed image sequence. Images of the left camera of the stereo rig at the edge of the airfield (left) and over the ground control station car next to the house (right).

6.1 Visual Odometry without and with Key Frames

In this pre-test, the relative movements $\mathbf{T}_{t_1:t_2}$ are integrated to absolute cartesian vehicle orientations $\mathbf{X}_{t_2}$ (position and rotation). Analogous to eq. 1, $\mathbf{X}$ are 4×4 matrices describing the vehicle position and attitude (direction cosine matrix) at the indexed time. This integration can be done if an initial $\mathbf{X}_0$ is available. The values of $\mathbf{X}_0$ are taken from the GNSS/INS flight state at the beginning of the image sequence. The set $\{\mathbf{X}_t : t = 1, \ldots, t_{\max}\}$ denotes now the path calculated only from accumulating image data.

In the incremental version, $\mathbf{X}_t$ is recursively

$$\begin{aligned} \mathbf{X}_t &= \mathbf{T}_t \cdot \mathbf{T}_{t-1} \cdot \ldots \cdot \mathbf{T}_1 \cdot \mathbf{X}_0 \\ \mathbf{X}_t &= \mathbf{T}_t \cdot \mathbf{X}_{t-1}, \end{aligned} \tag{14}$$

which means to accumulate a number of t relative movement steps. Contrary to that, the version with key frames calculates

$$\mathbf{X}_t = \mathbf{T}_{t-\tau:t} \cdot \mathbf{X}_{t-\tau}. \tag{15}$$

Depending of the size of $\tau - 1$ skipped frames within every accumulation step, much fewer relative movements have to be accumulated. This means theoretically, that the accumulation error is reduced and theoretically dissolved when the first frame can be kept as a key frame. This is the case when the current image is still overlapping with the first one, for example in hovering flight. However, the benefit from key frames should be decreasing when flying faster.

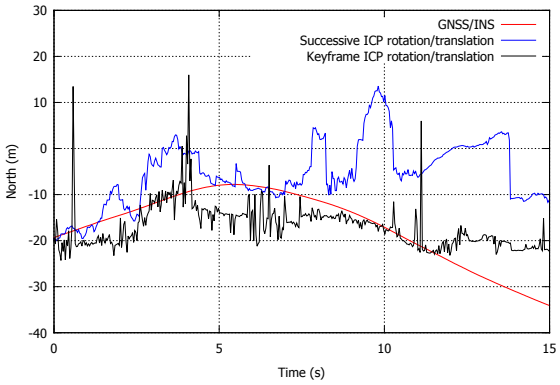

Fig. 8 Trajectory $\mathbf{X}_t$ by accumulating relative orientations from the Iterative Closest Point (ICP) algorithm, x-position coordinate. The black graph shows a successive accumulation of 150 image relations within 15 seconds. The blue graph shows the accumulation with key frames and fewer steps. Reference (red): GNSS/INS position.

As a result from a recorded image sequence, fig. 8 shows the integrated positions calculated from successive frames and with the use of key frames. Several trials have suggested that the optical flow of 150 to 250 features should be measured by the tracker for suitable results. The vision-based position starts without relative error to the GNSS/INS path and drifts due to translational and rotational errors with every update. The differences between both curves can be interpreted as follows: In the successive accumulation, the relative steps are quite small, and thus the errors (e.g. large jump at 13.5 s) are directly transferred to the next step (blue graph). Contrary to that, the relative steps are larger when referring to older key frames (black graph) and since the current step can cause a jump from one key frame to the next, some more fluctuation is transferred to the resulting positions. A positive effect is that erroneous steps are not integrated in every case, which is visible through the (removable) peaks (at 1 s, 4 s, 11 s) in the position coordinate. And the overall accumulation error is as expected lower than with successive relative accumulation. However, both methods accumulate errors so that a camera should not be used as the only sensor for navigation.

6.2 Visual Position Estimation with State Feedback

This next evaluation takes a longer image sequence and couples the flight state back to the visual motion estimation. Here, the GNSS/INS flight state is coupled back so that the resulting absolute position by visual relative measurements is not accumulating errors with regard to the GNSS/INS path. (Of course it is eventually drifting when the state estimator drifts in cases where no satellite signals are available.) This test only uses the key frame version of visual relative measurements. The relative transformation $\mathbf{T}_{t-\tau:t}$ is calculated by the transformed image point clouds into the geodetic system, i.e. the sets $\{\mathbf{q}^g_{t-\tau}\}$ and $\{\mathbf{q}^g_t\}$ with

$$\begin{aligned} \mathbf{q}^g_{t-\tau} &= \hat{\mathbf{X}}_{t-\tau} \cdot \mathbf{q}_{t-\tau}, \text{and} \\ \mathbf{q}^g_t &= \hat{\mathbf{X}}^-_t \cdot \mathbf{q}_t \end{aligned} \tag{16}$$

by using the transformation matrices containing the position and rotation of the corrected old state $\hat{\mathbf{X}}_{t-\tau}$ or the predicted current state $\hat{\mathbf{X}}^-_t$.

Results from flight tests are shown in the figures 9–11. The plots show excerpts from a 15-minute flight. It is examined whether a visual estimation of $\mathbf{T}_{t-\tau:t}$ with only three translational degrees of freedom (sec. 3.4) with the help of inertial rotations gives additional performance compared to the previous visual estimation of $\mathbf{T}_{t-\tau:t}$ with the full six degrees of freedom by using the ICP algorithm. Contrary to the vision-only method above, only 50 to 100 tracked features are required to get the viable results that are presented here. This decreases the computation time for image processing. For a better visualization where the state is fed back, a curve based on null transforms $\mathbf{T}_{t-\tau:t} = \mathbf{I}_{4\times4}$ is drawn into the plots. This results in horizontal lines with jumps to the GNSS/INS comparison plot every time a new key frame is used.

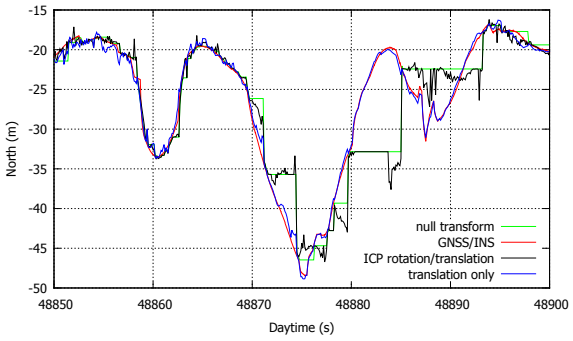

Fig. 9 Visual trajectory $\mathbf{X}_t$ by back-coupling the estimated state, x-coordinate. The plot includes visual 6-DoF estimation (black), 3-DoF estimation (blue), null transform (green), and GNSS/INS reference (red).

In fact, the figures do not prove a drift-free state estimation with only the visual odometry, but they indicate its behavior when used in combination with state feedback. The curves can be interpreted as the input of the vision-based positions into the navigation filter. The plots show that the visual 3-DoF estimation performs significantly better than the 6-DoF estimation, especially when the time between two key frames is large such as in the time between 48880 s and 48885 s, or 48885 s and

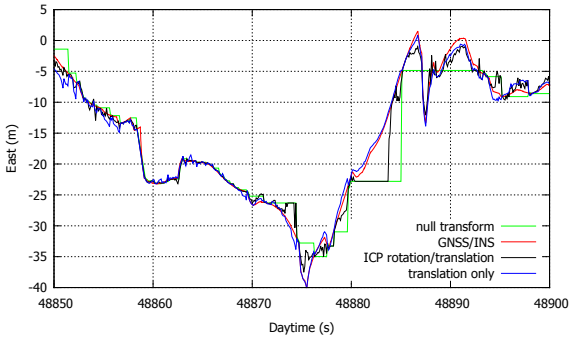

Fig. 10 Visual trajectory $\mathbf{X}_t$ by back-coupling the estimated state, y-coordinate.

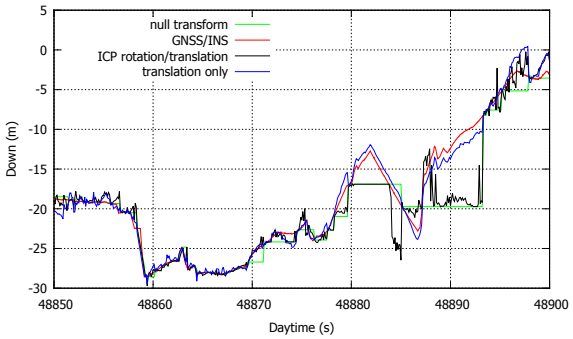

Fig. 11 Visual trajectory $\mathbf{X}_t$ by back-coupling the estimated state, z-coordinate.

48893 s. The higher performance of the 3-DoF estimation (translation only) is visible for all three axes. The most obvious reasons are scene geometry and texturing, and the rather narrow field of view of the cameras: The aircraft flies over a mostly planar scenery while looking downward, and therefore it is hard to distinguish between forward movements and pitch rotations, or between sideward movements and roll rotations from vision only. Here, the INS-based image feature point cloud transformation helps, and the 3-DoF estimation does not have to deal with these ambiguities. Beside that, it was observed that the highest uncertainty and errors are with the visual z-direction. This is mainly caused by the nature of stereo geometry with increasing range errors.

6.3 Integration of Visual Movements into the Flight State Estimator

The forward integration of stereo-based movements into the Kalman filter closes the loop between the navigation and image processing components. The following results are again based on the recorded data presented in the previous section. Here, the navigation EKF directly combines GPS pseudo-ranges, inertial data, and the visual 3-DoF movements that were improved with state predictions. With full satellite data reception, it was observed that the GNSS/INS trajectory is only slightly changed

when visual data are included. This is due to the state estimator that weights satellite data with a higher confidence.

To show the filter capabilities in the case of satellite signal dropouts, the path is calculated again from the raw data but with partly and fully disabled satellite data. Results are shown in the figures 12 to 14 where a visibility of less than four satellites is simulated. The evaluation is based on two (seconds 48850 to 48870) or three satellites (seconds 48870 to 48900). A further distinction between the reception of zero and incomplete satellite signals allows to show the effects of the GNSS raw data handling. Thus, the presented results give an idea how the filter handles incomplete satellite constellations.

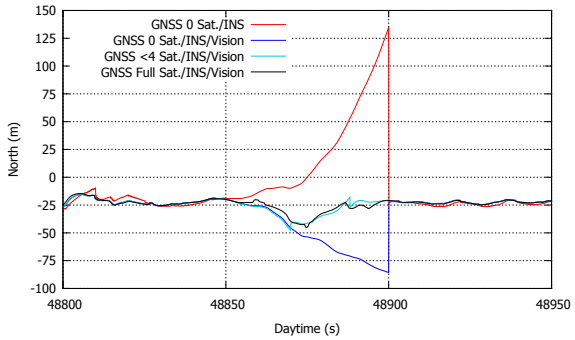

Fig. 12 Trajectory $\mathbf{X}_t$ from full flight state estimation, x-coordinate. GNSS signal dropout from 48850 s to 48900 s. Data from full dropout (0 satellites) with GNSS/INS only (red), full and partial dropout with visual 3-DoF estimation (dark and light blue), and complete data availability (black).

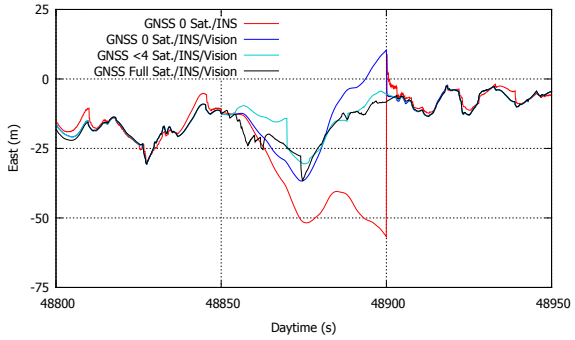

Fig. 13 Trajectory $\mathbf{X}_t$ from full flight state estimation, y-coordinate.

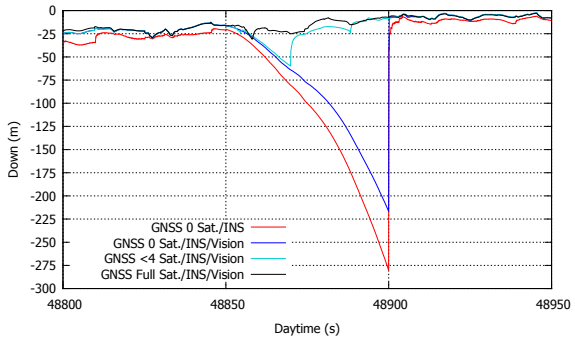

Fig. 14 Trajectory $\mathbf{X}_t$ from full flight state estimation, z-coordinate.

First of all, the results show that the inertial solution (red curves) has a high drift rate in all axes, its quadratic behavior is especially visible for the x- and the z-coordinates. After the 50 second dropout, errors of roughly 150 m (north), 50 m (east), and 250 m (down value) are observed with respect to a full satellite constellation (black curve). If now the visual odometry is switched on (dark blue curves), the errors can be reduced, but still remain large at 50 m (north), 25 m (east), and 200 m (down value). Especially the down value has the largest error, probably due to the nature of stereo cameras where the highest uncertainty is with the camera z-coordinate along the optical axis.

For partial satellite dropouts, large improvements are produced by the usage of GNSS pseudo-range data. The light blue curves show the estimated position in such a case. If at least two satellites are visible (seconds 48850 to 48870), the error is slightly reduced but still too large to navigate correctly. If three satellites are available (seconds 48870 to 48900), the error can be significantly reduced compared to the plot without satellites (dark blue), and no typical drift or other error accumulations are observed. With that, both visual odometry and satellite pseudo-range data evaluation improve the state estimation under these conditions. In these tests, at least three satellites are required to get a suitable state estimation.

7 Conclusion

Topic of this paper is the improvement of unmanned aircraft state estimation with the help of cameras. Satellite navigation and inertial data fusion is quite common, but comes with a lot of disadvantages like satellite signal errors especially in the proximity of obstacles and the high drift rates of small inertial measurement units which do not allow long integration times. However, this is the basis for the presented developments which use an extended Kalman filter solution for data fusion. The paper analyzes a variety of approaches how to measure the ego-motion with cameras by processing the image sequences. The presented option uses a stereo camera to handle the metric scaling issue that comes with monocular vision and computes the visible 3D movements within a sequence of such image pairs. The ego-motion can now be extracted from the characteristics of such visible movements, but it is often hard to decide whether the movements visible in the images are caused by the camera's rotation or by its displacement. To solve this ambiguity, the presented approach estimates the rotation by the inertial measurements so that the image processing part does only have to estimate the translational movement. It is shown with the tests that this easier estimation of only three degrees of freedom is significantly increasing the overall performance.

Similar to inertial data, visual information will accumulate errors over time because they are relative between two time stamps. Such errors can be reduced when the current image is still overlapping with a rather old one that has been stored as a key frame for movement determination, meaning a theoretical elimination of the accumulation error during hovering mode. In the presented tests, this is drifting also

at slower movements, and it has to be tested whether this will drift in hover as well. However, the combination of visual and inertial information is going to reduce the accumulation error when no satellite data is available, being an option for navigation with small unmanned vehicles.

Another aspect affects the absolute positioning with satellite data whose errors and dropout times should be reduced. Since many robotic applications directly use the position or velocity outputs from the receiver, no information is given when less than four satellites are visible. On the other hand, the known coupling methods of satellite pseudo-range with inertial data show that the positioning accuracy can be highly improved in cases where no full satellite solution is possible. Based on that, the presented filter also integrates the range data directly instead of using the pre-computed positions, being able to improve the overall performance in partly occluded areas.

Acknowledgment

The investigations are carried out within the project *Navigation zur Exploration von tiefffliegenden UAV in Katastrophenszenarien* (Navigation for exploration with low-flying UAVs in disaster scenarios, NExt UAV). The joint project is funded by the German Federal Ministry of Economics and Technology (BMWi) and administered by the space management of the German Aerospace Center (DLR) Bonn (support codes 50 NA 1002 and 50 NA 1003).

References

1. Achtelik, M., Achtelik, M., Weiss, S., Siegwart, R.: Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. In: IEEE International Conference on Robotics and Automation, pp. 3056–3063 (2012)
2. Adolf, F., et al.: An unmanned helicopter for autonomous flights in urban terrain. In: T. Kröger, F.M. Wahl (eds.) Advances in Robotics Research, pp. 275–285. Springer, Berlin (2009)
3. Ahrens, S., Levine, D., Andrews, G., How, J.P.: Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments. In: IEEE International Conference on Robotics and Automation, pp. 2643–2648 (2009)
4. Andert, F., et al.: Aerial tracking and GNSS-reference localization with a robotic total station. In: AIAA Infotech@Aerospace Conference (2012)
5. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-d point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence **9**(5), 698–700 (1987)
6. Barbour, N.M.: Inertial Navigation Sensors. In: NATO RTO Lecture Series, RTO-EN-SET-116, Low-Cost Navigation Sensors and Integration Technology (2011)
7. Becker, M., Bestmann, U., Sasse, A., Steen, M., Hecker, P.: In flight estimation of gyro and accelerometer scale factors for tactical and MEMS IMUs. In: ION GNSS 20th International Technical Meeting of the Satellite Division, pp. 2056–2065 (2007)
8. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(2), 239–256 (1992)
9. Bestmann, U., Steen, M., Becker, M., Sasse, A., Hecker, P.: Comparison of state and error state INS coupling filter based on real flight test data. In: ION GNSS 20th International Technical Meeting of the Satellite Division, pp. 2611–2618 (2007)

10. Carillo, L., López, A., Lozano, R., Pégard, C.: Combining stereo vision and inertial navigation system for a quad-rotor UAV. Journal of Intelligent and Robotic Systems **65**, 373–387 (2012)
11. Conte, G., Doherty, P.: A visual navigation system for UAS based on geo-referenced imagery. In: Conference on Unmanned Aerial Vehicle in Geomatics, UAV-g (2011)
12. Discher, C.: Ein dynamisches Fehlermodell für die Satellitenortung in einem integrierten INS/GNSS-Navigationssystem. Ph.D. thesis, Technische Universität Braunschweig (2003)
13. Eynard, D., Vasseur, P., Demonceaux, C., Frémont, V.: Real time UAV altitude, attitude and motion estimation from hybrid stereovision. Autonomous Robots **33**(1-2), 157–172 (2012)
14. Frietsch, N., et al.: Vision based hovering and landing system for a VTOL-MAV with geolocalization capabilities. In: AIAA Guidance, Navigation, and Control Conference (2008)
15. Grewal, M.S., Weill, L.R., Andrews, A.P.: Global Positioning Systems, Inertial Navigation, and Integration. John Wiley & Sons (2001)
16. Grießbach, D., Baumbach, D., Zuev, S.: Vision aided inertial navigation. In: ISPRS EuroCOW Conference (2010)
17. Hirschmüller, H., Innocent, P.R., Garibaldi, J.M.: Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics. In: International Conference on Control, Automation, Robotics and Vision, pp. 1099–1104 (2002)
18. Koch, A., Wittich, H., Thielecke, F.: A vision-based navigation algorithm for a VTOL UAV. In: AIAA Guidance, Navigation, and Control Conference and Exhibit (2006)
19. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
20. Matthies, L., Shafer, S.A.: Error modeling in stereo navigation. IEEE Journal of Robotics and Automation **3**(3), 239–248 (1987)
21. NGA: World geodetic system 1984 – its definition and relationships with local geodetic systems. Tech. rep., National Imagery and Mapping Agency (2000).
22. Nützi, G., Weiss, S., Scaramuzza, D., Siegwart, R.: Fusion of IMU and vision for absolute scale estimation in monocular SLAM. J. of Intell. and Robotic Systems **61**, 287–299 (2011)
23. Ramlall, R., et al.: Three satellite navigation in an urban canyon using a chip-scale atomic clock. In: ION GNSS Intl. Tech. Meeting of the Satellite Division, pp. 2937–2945 (2011)
24. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (2011). URL www.pointclouds.org
25. Sanfourche, M., et al.: Perception for UAV: Vision-based navigation and environment modeling. Aerospace Lab Journal **4** (2012)
26. SAPOS: Satellite positioning service of the German state survey. Website: http://www.sapos.de/
27. Weiss, S., Achtelik, M.W., Lynen, S., Chli, M., Siegwart, R.: Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In: IEEE International Conference on Robotics and Automation (2012)
28. Wendel, J., Trommer, G.F.: Tightly coupled GPS/INS integration for missile applications. Aerospace Science and Technology **8**, 627–634 (2004)
29. Winkler, S., Schulz, H., Buschmann, M., Kordes, T., Vörsmann, P.: Improving low-cost GPS/MEMS-based INS integration for autonomous MAV navigation by visual aiding. In: ION GNSS Intl. Tech. Meeting of the Satellite Division, pp. 1069–1075 (2004)