# Real-Time Mosaic-Aided Aerial Navigation: II. Sensor Fusion

Vadim Indelman [*] Pini Gurfil [†] Ehud Rivlin [‡]

*Technion - Israel Institute of Technology Haifa 32000, Israel*

Hector Rotstein [§]

*RAFAEL - Advanced Defense Systems Limited*

**We present a method for fusion of computer-vision mosaic-image-based motion estimation with a standard navigation system, yielding mosaic-aided navigation that does not rely on any a-priori information. The mosaic-based motion estimation uses inter-relations among images captured in real time during flight. This motion estimation is transformed into residual translation and rotation measurements, which are fed into a Kalman Filter, fusing the inertial measurement and the mosaic-based motion estimation. The proposed method can arrest and reduce the secular growth of inertial navigation errors, and correct measurements of the on-board inertial sensors. Moreover, we show that mosaic-aided navigation outperforms traditional vision-based aiding methods in challenging scenarios, such as flight over low-texture scenes captured by a camera with a narrow filed-of-view. To validate the proposed algorithms, we carried out a comprehensive performance evaluation, including statistical simulation runs and experiments based on real imagery.**

[*] Doctoral Student, Faculty of Aerospace Engineering. email: ivadim@tx.technion.ac.il

[†] Senior Lecturer, Faculty of Aerospace Engineering, Associate Fellow AIAA. email: pgurfil@technion.ac.il

[‡] Professor, Department of Computer Science. email: ehudr@cs.technion.ac.il

[§] Chief Control Systems Engineer. email: hector@rafael.co.il

# Nomenclature

| | |
|---|---|
| $\mathbf{b}$ | = accelerometer bias |
| $\mathbf{d}$ | = gyro drift |
| $H$ | = measurement matrix |
| $K$ | = Kalman filter gain matrix |
| $Q$ | = process noise covariance matrix |
| $R$ | = measurement noise covariance matrix |
| $\hat{R}_{C_1}^{C_2}$ | = image-based estimation of rotational motion |
| $T_B^A$ | = transformation matrix from system $A$ to system $B$ |
| $\hat{\mathbf{t}}_{1\to 2}$ | = image-based estimation of translational motion |
| $\mathbf{v}$ | = measurement noise vector |
| $\mathbf{w}$ | = process noise vector |
| $\mathbf{X}$ | = state vector |
| $\gamma$ | = scale constant |
| $\Delta\mathbf{P}$ | = position error vector |
| $\Delta\mathbf{V}$ | = velocity error vector |
| $\Delta\boldsymbol{\Psi}$ | = Euler angles error vector |
| $\Delta t$ | = time step |
| $\Phi$ | = roll angle |
| $\Phi_c$ | = continuous system matrix |
| $\Phi_d$ | = discrete system matrix |
| $\mu$ | = mean |
| $\Theta$ | = pitch angle |
| $\Psi$ | = yaw angle |
| $\boldsymbol{\Psi}$ | = Euler angles vector |
| $\sigma$ | = standard deviation |

*Subscripts*

| | |
|---|---|
| $B$ | = body coordinate system |
| $C$ | = camera coordinate system |
| $E$ | = Earth-fixed coordinate system |
| $L$ | = local-level, local-north coordinate system |
| $Nav$ | = computation based on the navigation system |
| $True$ | = true value |

# I. Introduction

In the first part of this work,[1] we suggested a method for mosaic-based motion estimation utilizing a scanning camera and a real-time mosaic construction. This paper focuses on fusion of this mosaic-based motion estimation with a standard navigation system of an airborne platform. The proposed method for mosaic-aided navigation does not rely on a digital terrain map (DTM) or the Global Positioning System (GPS); nor does it require geo-referencing of the mosaic image.

Vision-based algorithms use the information extracted from an image registration process, along with information available from other sensors, to improve the estimation of the platform's states and navigation parameters. For example, Ref. 4 proposed integrating the velocity-to-height vision-based estimation with additional on-board sensors; Ref. 5 applied the subspace-constraint approach[6] to partially estimate an airborne platform's states, based on measurements from an image registration process injected into an implicit extended Kalman filter (EKF); Ref. 7, showed that an EKF-based prediction allows better image registration, which, in turn, improves motion estimation; and Ref. 8 utilized epipolar constraints to aid the inertial navigation of a ground vehicle. All the preceding methods assumed that no *a-priori* information is available during the mission; we also adopt this assumption in the current work.

Various methods for improving navigation by fusing inertial information, vision information and observations from external sources have been proposed. Ref. 9 assumed that altimeter measurements are used for scaling the imaging sensors in order to improve state estimation during the landing phase of a space probe. Refs. 10 and 11 showed that absolute pose and motion estimation is possible when assuming that a DTM is available.

Another fusion approach is map-based navigation, which assumes that a map of the operational area is given and that the vehicle navigates by fusing inertial measurements, images of the environment and a map. In this context, Ref. 12 proposed a 2-phase position estimation: a) successive image registration and b) registration of the current image with a stored reference image. Ref. 13, 14 proposed vision-based underwater navigation for an unmanned underwater vehicle that relied on a previously-constructed mosaic image.

In contrast to the above, the vision-based navigation-aiding methods developed herein do not relay on any *a-priori* information; thus, the map of the overflown environment is not available but rather has to be constructed during the mission (the mosaic construction method implemented in this work is discussed in Part I of this work[1]).

Constructing a representation of environment while simultaneously enhancing an existing navigation system is known as *Simultaneous Localization and Mapping* (SLAM) (see e. g. Ref. 15 and references therein). The general approach for solving the SLAM problem is to augment the platform's state vector with parameters that describe the observed environment (e. g. features locations). However, the most conspicuous drawback of SLAM is computational load, which does not allow real-time performance once the augmented state has reached some critical size. Another difficulty is feature initialization. In this work, we assume that the mosaic image construction is an *independent* process, to be utilized for improving the performance of an existing navigation system. Therefore, our architecture alleviates the computational load required by SLAM. Moreover, we show how to use the camera scanning procedure and the mosaic construction process in order to improve vision-based navigation in difficult scenarios (narrow-FOV cameras, low-texture scenes). To the best of our knowledge, the latter aspect of mosaic-aided navigation has not been considered thus far.

The rest of this paper is organized as follows: Section II provides a measurement model development for fusing mosaic-based motion estimations with a standard navigation system. Section III elaborates the sensor-fusion algorithms. Section IV presents a performance evaluation of the proposed method for mosaic-aided navigation based on extensive experimentation; Section V concludes the discussion.

## II. Fusion of Image-Based Relative Motion Estimation with a Navigation System

Throughout this paper, the following coordinate systems are assumed (cf. Ref.[1]): $E$ is an Earth-fixed inertial reference frame, also known as an Earth-centered, Earth-fixed (ECEF) system; $L$ is a local-level, local-north (LLLN) system, also known as a north-east-down (NED) system; $B$ and $C$ are body-fixed and camera-fixed reference frames, respectively.

We assume that the relative motion parameters between each two image time instances, $t = t_1$ and t=$t_2$, were already extracted by the image processing module, regardless of whether the calculations had been performed by using the fundamental matrix or the homography matrix (cf. Ref. 1). Thus, the camera relative rotation matrix, $R_{C_1}^{C_2}$, transforming from the camera axes at time $t_2$ to the camera axes at time $t_1$, is known. In addition, the relative translation, $\mathbf{t}_{1\to2}^{C_2}$, is known up to some scale, $\gamma$. In the following discussion, we develop a measurement model, which relates the image-based estimated relative motion with the accumulating navigation errors of a standard inertial navigation system (INS).

Under *ideal conditions*, viz. when there are no navigation errors and $\mathbf{t}_{1\to2}^{C_2}$, $R_{C_1}^{C_2}$ are perfectly estimated, the following can be written:

$$\mathbf{Pos}_{True}^{L_2}(t_2) - \mathbf{Pos}_{True}^{L_2}(t_1) = \gamma T_{L_2,True}^{C_2}\mathbf{t}_{1\to2,True}^{C_2} \tag{1a}$$

$$T^{C_2}_{C_1,True} = R^{C_2}_{C_1,True} \tag{1b}$$

where $T^{C_2}_{L_2}$ is the directional cosines matrix (DCM) transforming from $C$ to LLLN at the time instance $t = t_2$; $T^{C_2}_{C_1}$ is the DCM transforming from $C$ at $t = t_2$ to $C$ at $t = t_1$; and $\mathbf{Pos}^{L(t_2)}(t_1)$ is the platform's position at $t = t_1$ expressed in the LLLN system at $t = t_2$, so that

$$\mathbf{Pos}^{L(t_2)}(t_1) = T^{L(t_1)}_{L(t_2)}\mathbf{Pos}^{L(t_1)}(t_1) \tag{2}$$

The subscript $(\cdot)_{True}$ in Eq. (1) indicates ideal conditions as defined above.

The DCM $T^{C_2}_{L_2}$ is required since the extracted translation $\mathbf{t}^{C_2}_{1\to 2}$ is given in the camera reference frame, while the left side of Eq. (1a) is expressed in the LLLN system.

Next, we will develop the measurement equations considering the navigation errors and the imperfect estimation of the relative motion. The analysis is performed based on the following state vector definition:

$$\mathbf{X} = \begin{bmatrix} \Delta\mathbf{P}^T & \Delta\mathbf{V}^T & \Delta\boldsymbol{\Psi}^T & \mathbf{d}^T & \mathbf{b}^T \end{bmatrix}^T \tag{3}$$

where $\Delta\mathbf{P} \in \mathbb{R}^3, \Delta\mathbf{V} \in \mathbb{R}^3, \Delta\boldsymbol{\Psi} \in SO(3)$ are the position, velocity and attitude errors, respectively, and $(\mathbf{d}, \mathbf{b})$ is the parametrization of errors in the inertial sensor measurements: $\mathbf{d} \in \mathbb{R}^3$ is the gyro drift, and $\mathbf{b} \in \mathbb{R}^3$ is the accelerometer bias. The first 9 components of $\mathbf{X}$ are given in LLLN coordinates, while the last 6 are written in a body-fixed reference frame.

### A. Translation Measurement Equation

In an ideal situation, with no navigation and image processing errors, the two sides of Eq. (1a) constitute parallel vectors. Thus, this equation yields the following constraint:

$$\left[\mathbf{Pos}^{L_2}_{True}(t_2) - \mathbf{Pos}^{L_2}_{True}(t_1)\right] \times T^{C_2}_{L_2,True}\mathbf{t}^{C_2}_{1\to 2,True} = \mathbf{0} \tag{4}$$

In reality, there are navigation errors that increase with time (Appendix B). Moreover, the estimated camera matrix contains errors due to image noise. Thus, Eq. (4) no longer holds. Denoting by $Nav$ parameters that are taken from the navigation data and by $\hat{\mathbf{t}}^{C_2}_{1\to 2}$ the actual estimated translation vector obtained from the image processing module, Eq. (4) becomes

$$\left[\mathbf{Pos}^{L_2}_{Nav}(t_2) - \mathbf{Pos}^{L_2}_{Nav}(t_1)\right] \times T^{C_2}_{L_2,Nav}\hat{\mathbf{t}}^{C_2}_{1\to 2} = \mathbf{z}_{translation} \tag{5}$$

where $\mathbf{z}_{translation}$ denotes the residual measurement vector.

Taking into account the fact that $\mathbf{Pos}^{L_2}_{Nav}(.) = \mathbf{Pos}^{L_2}_{True}(.) + \Delta\mathbf{P}^{L_2}(.)$ and subtracting (4) from (5) results in

$$\begin{aligned} &[\mathbf{Pos}_{True}(t_2) - \mathbf{Pos}_{True}(t_1) + \Delta\mathbf{P}(t_2) - \Delta\mathbf{P}(t_1)]^{L_2} \times T^{C_2}_{L_2,Nav}\hat{\mathbf{t}}^{C_2}_{1\to 2} - \\ &\qquad [\mathbf{Pos}_{True}(t_2) - \mathbf{Pos}_{True}(t_1)]^{L_2} \times T^{C_2}_{L_2,True}\mathbf{t}^{C_2}_{1\to 2,True} = \mathbf{z}_{translation} \end{aligned} \tag{6}$$

or

$$[\Delta\mathbf{P}(t_2) - \Delta\mathbf{P}(t_1)]^{L_2} \times T^{C_2}_{L_2,Nav}\hat{\mathbf{t}}^{C_2}_{1\to 2} + \mathbf{v} = \mathbf{z}_{translation} \tag{7}$$

where $\mathbf{v} = [\mathbf{Pos}_{True}(t_2) - \mathbf{Pos}_{True}(t_1)]^{L_2} \times \left[T^{C_2}_{L_2,Nav}\hat{\mathbf{t}}^{C_2}_{1\to 2} - T^{C_2}_{L_2,True}\mathbf{t}^{C_2}_{1\to 2,True}\right]$. The vector $\mathbf{v}$ is due to imperfect translation measurements and navigation errors. One may verify that in ideal conditions this term is nullified.

According to Appendix B (Eq. (56)), the inertial position error for a sufficiently small $\Delta t = t_2 - t_1$ or for a straight and level flight is given by (the $Nav$ subscript is omitted for simplicity from here on; thus, all parameters are computed based on the navigation system data, unless otherwise specified):

$$\begin{aligned} \Delta\mathbf{P}(t_2) &= T^{L_1}_{L_2}\Big[-\frac{1}{6}A_s(t_1)T^{B_1}_{L_1}\mathbf{d}\cdot(\Delta t)^3 \\ &+ \frac{1}{2}\left[A_s(t_1)\Delta\boldsymbol{\Psi}(t_1) + T^{B_1}_{L_1}\mathbf{b}\right](\Delta t)^2 + \Delta\mathbf{V}(t_1)\Delta t + \Delta\mathbf{P}(t_1)\Big] \end{aligned} \tag{8}$$

Note that a transformation matrix, $T^{L_1}_{L_2}$, was added to express the position error at $t = t_2$ in LLLN coordinates.

Substituting Eq. (8) into Eq. (7), canceling position errors at $t = t_1$ and denoting $\hat{\mathbf{t}}^{L_2}_{1\to 2} \equiv T^{C_2}_{L_2,Nav}\hat{\mathbf{t}}^{C_2}_{1\to 2}$ yields

$$\begin{aligned} &\left\{T^{L_1}_{L_2}\left[-\frac{1}{6}A_s(t_1)T^{B_1}_{L_1}\mathbf{d}\cdot(\Delta t)^3 + \frac{1}{2}\left[A_s(t_1)\Delta\boldsymbol{\Psi}(t_1) + T^{B_1}_{L_1}\mathbf{b}\right](\Delta t)^2 + \Delta\mathbf{V}(t_1)\Delta t\right]\right\} \times \hat{\mathbf{t}}^{L_2}_{1\to 2} \\ &\qquad + \mathbf{v} = \mathbf{z}_{translation} \end{aligned} \tag{9}$$

We shall now use the wedge operator, defined for some vector $\boldsymbol{\xi} = [\xi_1, \xi_2, \xi_3]^T$ as

$$\boldsymbol{\xi}^{\wedge} = \begin{bmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{bmatrix} \tag{10}$$

Thus, Eq. (9) becomes

$$\left[T_{L_2}^{L_1}\left[-\frac{1}{6}A_s(t_1)T_{L_1}^{B_1}\mathbf{d}\cdot(\Delta t)^3+\frac{1}{2}\left[A_s(t_1)\Delta\mathbf{\Psi}(t_1)+T_{L_1}^{B_1}\mathbf{b}\right](\Delta t)^2+\Delta\mathbf{V}(t_1)\Delta t\right]\right]^{\wedge}\hat{\mathbf{t}}_{1\to2}^{L_2}$$
$$+\mathbf{v}=\mathbf{z}_{translation} \quad (11)$$

One can see from Eq. (11) that the translation measurement equation is of the form $\mathbf{z}_{translation} = H^{tr}\mathbf{X}+\mathbf{v}$, where

$$H^{tr}=\begin{bmatrix}0_{3\times3} & H_{\Delta V}^{tr} & H_{\Delta\Psi}^{tr} & H_{d}^{tr} & H_{b}^{tr}\end{bmatrix} \quad (12)$$

After some algebraic manipulations (cf. Appendix A), the submatrices of $H^{tr}$ can be rendered into

$$H_{\Delta V}^{tr} = -\left[\hat{\mathbf{t}}_{1\to2}^{L_2}\right]^{\wedge}T_{L_2}^{L_1}\Delta t \quad (13a)$$
$$H_{\Delta\Psi}^{tr} = -\frac{1}{2}\left[\hat{\mathbf{t}}_{1\to2}^{L_2}\right]^{\wedge}T_{L_2}^{L_1}A_s(t_1)(\Delta t)^2 \quad (13b)$$
$$H_{d}^{tr} = \frac{1}{6}\left[\hat{\mathbf{t}}_{1\to2}^{L_2}\right]^{\wedge}T_{L_2}^{L_1}A_s(t_1)T_{L_1}^{B_1}(\Delta t)^3 \quad (13c)$$
$$H_{b}^{tr} = -\frac{1}{2}\left[\hat{\mathbf{t}}_{1\to2}^{L_2}\right]^{\wedge}T_{L_2}^{L_1}T_{L_1}^{B_1}(\Delta t)^2 \quad (13d)$$

### B. Rotation Measurement Equation

Recall Eq. (1b), written under the assumption of ideal conditions: $T_{C_1,True}^{C_2} = R_{C_1,True}^{C_2}$. When taking into account navigation errors and errors in the estimated rotation matrix, this equation no longer holds. Instead, we define a residual error angle vector, $\mathbf{z}_{Rotation}$. Under the assumption of small angles, this vector can be written as

$$I-\mathbf{z}_{rotation,A}^{\wedge}=T_{C_1,Nav}^{C_2}\left[\hat{R}_{C_1}^{C_2}\right]^{T} \quad (14)$$

Here $T_{C_1,Nav}^{C_2}$ denotes the DCM transforming from $C$ at $t=t_1$ to $C$ at $t=t_2$, computed by the navigation system of the platform. This matrix differs from the true DCM due to platform navigation errors. The matrix $\hat{R}_{C_1}^{C_2}$ is the estimated rotation matrix. One can verify that under ideal conditions, $T_{C_1,True}^{C_2} = R_{C_1,True}^{C_2}$, and thus the rotation error angle $\mathbf{z}_{Rotation}$ is equal to zero. We omit the subscript $(Nav)$ for simplicity, and write simply $T_{C_1}^{C_2}$.

In general, $T_{C_1}^{C_2}$ can be written as follows:

$$T_{C_1}^{C_2}=\underbrace{T_{C_1}^{B_1}T_{B_1}^{L_1}T_{L_1}^{E}}_{T_{C_1}^{E}}\underbrace{T_{E}^{L_2}T_{L_2}^{B_2}T_{B_2}^{C_2}}_{T_{E}^{C_2}} \quad (15)$$

where the matrices $T_{C_1}^{B_1}$ and $T_{B_2}^{C_2}$ are assumed to be known precisely – or at least with much more precision compared to the developing attitude errors. Thus, $T_B^C = T_{B,True}^C$. The errors in the ECEF to LLLN rotation matrix are assumed to be negligible, since they depend on the position errors, which are small relative to Earth's radius. Thus, $T_L^E = T_{L,True}^E$.

In addition to position errors, the platform navigation system tends to develop attitude errors. These errors do not allow a perfect estimation of the DCM transforming from LLLN to $B$, since the estimated LLLN system is erroneous. Thus, there are two LLLN systems: The *true* LLLN system, denoted by $L$, and the platform's estimation of the LLLN system, denoted by $L_C$. Taking this into consideration, the platform's *believed* DCM, $T_B^L$, can be written as

$$T_B^L = T_B^{L_C} T_{L_C}^L \tag{16}$$

where $T_{L_C}^L$ is the DCM transforming between the two LLLN systems. Assuming small attitude errors, we write $\mathbf{\Psi}_{Nav} = \mathbf{\Psi}_{True} + \Delta\mathbf{\Psi}$ to obtain

$$T_{L_C}^L = I - \Delta\mathbf{\Psi}^\wedge \tag{17}$$

Based on Eq. (54) from Appendix B, for a sufficiently small $t - t_0$ or for a straight and level flight, one can use the approximation

$$\Delta\mathbf{\Psi}(t) = -T_L^B(t_0)\mathbf{d}\Delta t + \Delta\mathbf{\Psi}(t_0) \tag{18}$$

Under this approximation, the DCM transforming from $L$ to $L_C$ assumes the following form:

$$\begin{aligned} T_{L_C}^L(t_1) &= I - \Delta\mathbf{\Psi}^\wedge(t_1) && \text{(19a)} \\ T_{L_C}^L(t_2) &= I - \left[\Delta\mathbf{\Psi}(t_1) - T_{L_1}^{B_1}\mathbf{d}\Delta t\right]^\wedge && \text{(19b)} \end{aligned}$$

Thus, the matrices $T_{B_1}^{L_1}$ and $T_{L_2}^{B_2}$ in Eq. (15) can be written as

$$\begin{aligned} T_{B_1}^{L_1} &= T_{B_1}^{L_{C_1}}\left[I - \Delta\mathbf{\Psi}^\wedge(t_1)\right] && \text{(20a)} \\ T_{L_2}^{B_2} &= \left[I + \left(\Delta\mathbf{\Psi}(t_1) - T_{L_1}^{B_1}\mathbf{d}\Delta t\right)^\wedge\right] T_{L_{C_2}}^{B_2} && \text{(20b)} \end{aligned}$$

Substituting Eqs. (20) into Eq. (15) yields

$$\begin{aligned} T_{C_1}^{C_2} &= T_{C_1}^{B_1} T_{B_1}^{L_{C_1}} \left\{ T_{L_1}^E T_E^{L_2} \left[I + \left(\Delta\mathbf{\Psi}(t_1) - T_{L_1}^{B_1}\mathbf{d}\Delta t\right)^\wedge\right] - \Delta\mathbf{\Psi}^\wedge(t_1) T_{L_1}^E T_E^{L_2} \right\} T_{L_{C_2}}^{B_2} T_{B_2}^{C_2} \\ &+ O\left(\Delta\mathbf{\Psi}^2(t_1), \Delta\mathbf{\Psi}(t_1)\mathbf{d}\right) && \text{(21)} \end{aligned}$$

After ignoring second-order terms and carrying out some additional algebraic manipulations we get

$$
\begin{aligned}
T_{C_1}^{C_2} &= T_{C_1}^{B_1} T_{B_1}^{L_{C_1}} T_{L_1}^{E} T_{E}^{L_2} T_{L_{C_2}}^{B_2} T_{B_2}^{C_2} \\
&+ T_{C_1}^{B_1} T_{B_1}^{L_{C_1}} \left[ T_{L_1}^{E} T_{E}^{L_2} \left( \Delta\boldsymbol{\Psi}(t_1) - T_{L_1}^{B_1} \mathbf{d} \Delta t \right)^{\wedge} - \Delta\boldsymbol{\Psi}^{\wedge}(t_1) T_{L_1}^{E} T_{E}^{L_2} \right] T_{L_{C_2}}^{B_2} T_{B_2}^{C_2} \qquad (22)
\end{aligned}
$$

Note that $T_{C_1}^{B_1} T_{B_1}^{L_{C_1}} T_{L_1}^{E} T_{E}^{L_2} T_{L_{C_2}}^{B_2} T_{B_2}^{C_2}$ is the nominal value of $T_{C_1,True}^{C_2}$ and thus

$$
T_{C_1}^{B_1} T_{B_1}^{L_{C_1}} T_{L_1}^{E} T_{E}^{L_2} T_{L_{C_2}}^{B_2} T_{B_2}^{C_2} = R_{C_1,True}^{C_2} \qquad (23)
$$

As was mentioned before, the rotation matrix that was estimated by the image processing module differs from the true matrix. Let $T_{R_{Err}}$ be the DCM transforming between the true rotation matrix and the estimated one: $\hat{R}_{C_1}^{C_2} = T_{R_{Err}} R_{C_1,True}^{C_2}$.

Multiplying Eq. (22) by $\left[ \hat{R}_{C_1}^{C_2} \right]^T$ from the right and using Eq. (23) yields

$$
\begin{aligned}
T_{C_1}^{C_2} \left[ \hat{R}_{C_1}^{C_2} \right]^T &= \Big\{ I + T_{C_1}^{B_1} T_{B_1}^{L_{C_1}} \Big[ T_{L_1}^{E} T_{E}^{L_2} \left( \Delta\boldsymbol{\Psi}(t_1) - T_{L_1}^{B_1} \mathbf{d} \Delta t \right)^{\wedge} \\
&- \Delta\boldsymbol{\Psi}^{\wedge}(t_1) T_{L_1}^{E} T_{E}^{L_2} \Big] T_{L_{C_2}}^{B_2} T_{B_2}^{C_2} R_{C_1,True}^{C_2} \Big\} T_{R_{err}}^{T} \qquad (24)
\end{aligned}
$$

Substituting Eq. (24) into Eq. (14) yields

$$
\begin{aligned}
z_{rotation}^{\wedge} = \left[ I - T_{R_{err}}^{T} \right] + T_{C_1}^{B_1} T_{B_1}^{L_{C_1}} \Big[ &- T_{L_1}^{E} T_{E}^{L_2} \left( \Delta\boldsymbol{\Psi}(t_1) - T_{L_1}^{B_1} \mathbf{d} \Delta t \right)^{\wedge} \\
&+ \Delta\boldsymbol{\Psi}^{\wedge}(t_1) T_{L_1}^{E} T_{E}^{L_2} \Big] T_{L_{C_2}}^{B_2} T_{B_2}^{C_2} \left[ \hat{R}_{C_1}^{C_2} \right]^T \qquad (25)
\end{aligned}
$$

Assuming small estimation rotation errors $\mathbf{v}_R$, one can write $T_{R_{err}}^{T} = I - \mathbf{v}_R^{\wedge}$. Thus, the preceding equation becomes

$$
\begin{aligned}
z_{rotation}^{\wedge} = \mathbf{v}_R^{\wedge} + T_{C_1}^{B_1} T_{B_1}^{L_{C_1}} \Big[ &- T_{L_1}^{E} T_{E}^{L_2} \left( \Delta\boldsymbol{\Psi}(t_1) - T_{L_1}^{B_1} \mathbf{d} \Delta t \right)^{\wedge} \\
&+ \Delta\boldsymbol{\Psi}^{\wedge}(t_1) T_{L_1}^{E} T_{E}^{L_2} \Big] T_{L_{C_2}}^{B_2} T_{B_2}^{C_2} \left[ \hat{R}_{C_1}^{C_2} \right]^T \qquad (26)
\end{aligned}
$$

Using Eq. (15), one can write the following two relations:

$$
T_{C_1}^{B_1} T_{B_1}^{L_{C_1}} T_{L_1}^{E} T_{E}^{L_2} = \hat{R}_{C_1}^{C_2} T_{C_2}^{B_2} T_{B_2}^{L_{C_2}} \qquad (27a)
$$

$$
T_{C_1}^{B_1} T_{B_1}^{L_{C_1}} = \hat{R}_{C_1}^{C_2} T_{C_2}^{B_2} T_{B_2}^{L_{C_2}} T_{L_2}^{E} T_{E}^{L_1} \qquad (27b)
$$

Substituting Eqs. (27) into Eq. (26) entails

$$\begin{aligned} z^{\wedge}_{rotation} = -\hat{R}^{C_2}_{C_1} T^{B_2}_{C_2} T^{L_{C_2}}_{B_2} \left( \Delta\mathbf{\Psi}(t_1) - T^{B_1}_{L_1} \mathbf{d}\Delta t \right)^{\wedge} T^{B_2}_{L_{C_2}} T^{C_2}_{B_2} \left[ \hat{R}^{C_2}_{C_1} \right]^T \\ + \hat{R}^{C_2}_{C_1} T^{B_2}_{C_2} T^{L_{C_2}}_{B_2} T^{E}_{L_2} T^{L_1}_{E} \Delta\mathbf{\Psi}^{\wedge}(t_1) T^{E}_{L_1} T^{L_2}_{E} T^{B_2}_{L_{C_2}} T^{C_2}_{B_2} \left[ \hat{R}^{C_2}_{C_1} \right]^T + \mathbf{v}^{\wedge}_{R} \end{aligned} \quad (28)$$

Using the fact that for any matrix $\Lambda$ and any vector $\boldsymbol{\xi}$, $\Lambda \boldsymbol{\xi}^{\wedge} \Lambda^T = (\Lambda \boldsymbol{\xi})^{\wedge}$, the above expression transforms into

$$z^{\wedge}_{rotation} = \left[ \hat{R}^{C_2}_{C_1} T^{B_2}_{C_2} T^{L_{C_2}}_{B_2} T^{E}_{L_2} T^{L_1}_{E} \Delta\mathbf{\Psi}(t_1) \right]^{\wedge} - \left[ \hat{R}^{C_2}_{C_1} T^{B_2}_{C_2} T^{L_{C_2}}_{B_2} \left( \Delta\mathbf{\Psi}(t_1) - T^{B_1}_{L_1} \mathbf{d}\Delta t \right) \right]^{\wedge} + \mathbf{v}^{\wedge}_{R} \quad (29)$$

Thus

$$z_{rotation} = R_{1-2} T^{B_2}_{C_2} T^{L_{C_2}}_{B_2} \left( T^{E}_{L_2} T^{L_1}_{E} - I \right) \Delta\mathbf{\Psi}(t_1) + R_{1-2} T^{B_2}_{C_2} T^{L_{C_2}}_{B_2} T^{B_1}_{L_1} \mathbf{d}\Delta t + \mathbf{v}_R \quad (30)$$

One can see that Eq. (30) is of the form $\mathbf{z}_{rotation} = H^{rot}\mathbf{X} + \mathbf{v}_R$, where

$$H^{rot} = \left[ 0_{3\times3} \quad 0_{3\times3} \quad H^{rot}_{\Delta\Psi} \quad H^{rot}_{d} \quad 0_{3\times3} \right] \quad (31a)$$

$$H^{rot}_{\Delta\Psi} = \hat{R}^{C_2}_{C_1} T^{B_2}_{C_2} T^{L_{C_2}}_{B_2} \left( T^{E}_{L_2} T^{L_1}_{E} - I \right) \quad (31b)$$

$$H^{rot}_{d} = \hat{R}^{C_2}_{C_1} T^{B_2}_{C_2} T^{L_{C_2}}_{B_2} T^{B_1}_{L_1} \Delta t \quad (31c)$$

# III. Mosaic-Based Aiding and Sensor Fusion Algorithms

## A. Main Constituents

The mosaic-based navigation aiding system consists of three main modules (cf. Figure 1): A navigation module, a camera scanning module and an image processing module.

The navigation phase consists of the following steps: (a) Trajectory generation; (b) velocity and angular increments extraction from the created trajectory; (c) inertial measurement unit (IMU) error definition and contamination of pure increments by noise; and (d) strapdown calculations. The strapdown mechanism provides, at each time step, the calculated position, velocity and attitude of the platform. In parallel to the strapdown calculations, at a much slower rate, Kalman filter calculations are performed based on the available measurements. At the end of each filter cycle, the strapdown output is updated with the estimated state vector of the filter.

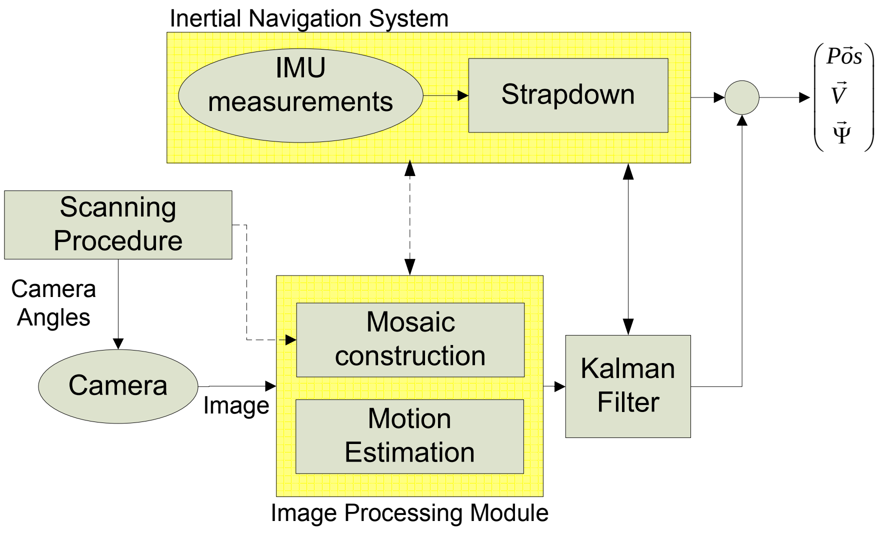


**Figure 1. Overview of the system concept.**

As described in Ref. 1, the camera scanning module provides camera angle commands that yield a continuous scan, in which each image overlaps with a preceding image, as well as with an image from a previous scan stripe (see Figure 4 in Ref. 1).

The image processing algorithm performs relative motion estimation based on previous and current images. The previous image can be either the mosaic image, or an image that was last captured. The algorithm is also capable of calculating *ideal* relative motion measurements based on the platform trajectory, without actually using any real images. These ideal pseudo-measurements were used as baseline for evaluating the best possible performance of the proposed method.

## B. Sensor Fusion

The image-based motion estimation is fused with INS data using a Kalman Filter (KF) applied to the measurement model developed in Section II. The estimated state vector $\hat{\mathbf{X}}$ is used for correcting the navigation solution from the INS and for correcting the IMU measurements using the estimated bias and drift values (cf. Figure 1). After each such update step, the estimated state vector is nullified, i. e., the *a-posteriori* estimation at some time instant $t_k$, $\hat{\mathbf{X}}_{k|k}$, is set to $\mathbf{0}$.

Next, we provide some details on the Kalman Filter equations implemented in this work. The propagation step involves computation of an *a-priori* covariance matrix $P_{k+1|k}$ according to

$$P_{k+1|k} = \Phi_d(k+1,k)P_{k|k}\Phi_d^T(k+1,k) + Q_k \tag{32}$$

where $\Phi_d(k+1,k), P_{k|k}, Q_k$ are the process discrete system matrix, *a-posteriori* covariance matrix, and the process noise covariance matrix, respectively. The discrete system matrix $\Phi_d$ is computed based on the continuous system matrix $\Phi_c$, defined in Eq. (52), using

$$\Phi_d = e^{\Phi_c \Delta t} \tag{33}$$

where $\Phi_c$ is evaluated based on data taken from the navigation system, and $\Delta t$ is the propagation step time interval. The process noise covariance $Q_k$ is a constant matrix, whose entries are determined by filter tuning.

Note that the propagation step does *not* include propagation of the state vector based on the *a-posteriori* estimation of the state vector from the previous time step, since, as mentioned above, $\hat{\mathbf{X}}_{k|k}$ is nullified immediately after being computed. Thus, $\hat{\mathbf{X}}_{k+1|k} = \mathbf{0}$.

After performing the propagation step, an update step is engaged given that the motion estimation $(\mathbf{t}_{1\to 2}^{C_2}, R_{C_1}^{C_2})$ is received from the image processing module. First, the Kalman filter gain matrix is computed according to

$$K_{k+1} = P_{k+1|k} H_{k+1}^T \left[ H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1} \right]^{-1} \tag{34}$$

where

$$H_{k+1} = \begin{bmatrix} H_{k+1}^{tr} \\ H_{k+1}^{rot} \end{bmatrix} \tag{35}$$

Here $H^{tr}, H^{rot}$ are the translation and rotation measurements matrices defined in Eqs. (12) and (31), respectively. The matrix $R_{k+1}$ in Eq. (34) is a measurement noise covariance matrix, which is of the following block-diagonal form

$$R_{k+1} = \begin{bmatrix} R_{k+1}^{tr} & 0_{3\times 3} \\ 0_{3\times 3} & R_{k+1}^{rot} \end{bmatrix} \tag{36}$$

where $R^{tr}, R^{rot}$ are the translation and rotation measurements noise covariance matrices, respectively. While $R^{rot}$ is a constant matrix, an adaptive translation measurement noise covariance matrix $R^{tr}$ is assumed due to Eq. (7):

$$R^{tr} = -\left[ \mathbf{Pos}_{Nav}^{L_2}(t_2) - \mathbf{Pos}_{Nav}^{L_2}(t_1) \right]^{\wedge} R_{est} \left[ \mathbf{Pos}_{Nav}^{L_2}(t_2) - \mathbf{Pos}_{Nav}^{L_2}(t_1) \right]^{\wedge} \tag{37}$$

where $R_{est}$ is a $3 \times 3$ tuning matrix that represents the level of accuracy in the vision-based estimation of the translation direction. For example, in the experiments with real imagery presented in Section IV-C, it was assumed to be close to $I_{3\times 3}$.

Once the Kalman Filter gain matrix is available, *a-posteriori* values of the state vector and covariance matrix are computed as

$$\begin{aligned}
\hat{\mathbf{X}}_{k+1|k+1} &= K_{k+1}\mathbf{Z}_{k+1} && (38)\\
P_{k+1|k+1} &= \left[I - K_{k+1}H_{k+1}\right]P_{k+1|k}\left[I - K_{k+1}H_{k+1}\right]^T + K_{k+1}R_{k+1}K_{k+1}^T && (39)
\end{aligned}$$

where

$$\mathbf{Z} \equiv \begin{bmatrix} \mathbf{z}_{translation} \\ \mathbf{z}_{rotation} \end{bmatrix} \tag{40}$$

and $\mathbf{z}_{translation}$, $\mathbf{z}_{rotation}$ are the residual measurements, computed according to Eqs. (5) and (14), respectively. The equation for the *a-posteriori* state vector, given in Eq. (38), is a degeneration of the standard equation $\hat{\mathbf{X}}_{k+1|k+1} = \hat{\mathbf{X}}_{k+1|k} + K_{k+1}\left(\mathbf{Z}_{k+1} - H_{k+1}\hat{\mathbf{X}}_{k+1|k}\right)$, since $\hat{\mathbf{X}}_{k+1|k} = \mathbf{0}$.

The imperfectness of the image-based relative motion estimation is projected onto the unobservable states, resulting in degraded estimation of these states. In order to reduce this undesired phenomenon, a fictitious ideal velocity measurement is used in addition to the relative motion measurements, so that

$$(\mathbf{V}^L_{true})^T\Delta\mathbf{V} = \mathbf{0}, \tag{41}$$

implying nullification of the velocity error along the flight heading. The term $\mathbf{V}^L_{true}$ refers to the true value of the platform velocity in the LLLN system. Since it is unknown, the platform velocity $\mathbf{V}^L$ taken from the navigation system, is used instead.

A Kalman filter gain matrix, $K$, is computed according to Eq. (34) based on an *a-priori* covariance matrix $P_{k+1|k}$, an augmented measurement matrix, $H_{aug} = [H^T, H_v^T]^T$, and an augmented measurement noise covariance matrix, $R_{aug}$, where

$$H_v = \begin{bmatrix} 0_{1\times3} & (\mathbf{V}^L)^T & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} \end{bmatrix} \tag{42}$$

and $H$ is the measurement matrix given in Eq. (35).

The augmented measurement noise covariance matrix $R_{aug}$ is of the following form

$$R_{aug} = \begin{bmatrix} R & 0 \\ 0_{1\times3} & R_v \end{bmatrix} \tag{43}$$

where $R$ is given in Eq. (37) and $R_v$ is the fictitious velocity measurement noise covariance matrix, which constitutes a tuning parameter. Small-valued entries in $R_v$ indicate that this additional measurement is reliable, and therefore other measurements will have a minor influence on the entries of the gain matrix $K$, corresponding to position and velocity along the flight heading. This, in turn, prevents from erroneous image-based relative motion measurements to affect the unobservable states.

Once $K$ is computed, its last column is discarded, since it represents the fictitious velocity measurement. The time update step of the state and the covariance matrix is performed without using any additional information. In addition, due to the varying quality of the image measurements (see Ref. 1), a measurements-rejection mechanism is used to avoid fusion of low-quality measurements (outliers).

# IV. Experimental Results for Mosaic-Aided Navigation

This section contains simulation results of the developed mosaic-aided navigation method. The majority of the experiments are based on real image sequences acquired using Google Earth (cf. Section IV-B in Part I of this work[1]). We note that the simulation runs were performed without the captive flight stage, during which the platform partially estimates its IMU errors from an external reference system.

The assumed $1\sigma$ values of IMU errors and initial navigation errors are given in Table 1. Actual values of initial navigation errors and IMU errors in the statistical simulation runs are determined by drawing samples from a zero-mean normal distribution with a standard deviation $\sigma$, that is, the value of some parameter $s_i$ is drawn according to $s_i \sim N(0, \sigma_{s_i})$.

**Table 1. Initial Navigation Errors and IMU Errors**

| Parameter | Description | Value | Units |
|---|---|---|---|
| $\Delta\mathbf{P}$ | Initial position error ($1\sigma$) | $(100, 100, 100)^T$ | m |
| $\Delta\mathbf{V}$ | Initial velocity error ($1\sigma$) | $(0.3, 0.3, 0.3)^T$ | m/s |
| $\Delta\mathbf{\Psi}$ | Initial attitude error ($1\sigma$) | $(0.1, 0.1, 0.1)^T$ | deg |
| $\mathbf{d}$ | IMU drift ($1\sigma$) | $(1, 1, 1)^T$ | deg/hr |
| $\mathbf{b}$ | IMU bias ($1\sigma$) | $(1, 1, 1)^T$ | mg |

## A. Ideal Pseudo-Measurements

We start by presenting the performance of the proposed method assuming *ideal* image-based relative motion measurements. This is the best possible performance, since the real-world image-based measurements are imperfect. The ideal image-based relative motion measurements, which are referred to *pseudo-measurements*, were calculated based on the platform true trajectory.

Figures 2-6 show Monte-Carlo results for a straight and level north-heading trajectory, in which the ideal pseudo-measurements were injected into a Kalman filter at a 1 Hz frequency. Each figure contains 5 curves: mean ($\mu$), mean+standard deviation ($\mu + \sigma$), minimum and maximum values, and the square root of the filter covariance, defined for the $i$-th component in the state vector $\mathbf{X}$ as $\sqrt{P(i,i)}$, where $P$ is the *a-posteriori* covariance matrix.

Velocity errors (Figure 3) normal to the flight heading are reduced and blocked. However, due to errors introduced by expressing the translation measurement in the LLLN system, these errors are not nullified. As a consequence, position errors (Figure 2) normal to the flight heading are considerably reduced compared to an inertial scenario. Velocity errors and position errors along the flight heading are not diminished due to lack of observability.

The roll angle error $\Delta\Phi$ (Figure 4) is significantly reduced (from an initial error of 0.1 deg to a stable 0.05 deg error), while pitch and yaw angles errors $(\Delta\Theta, \Delta\Psi)$, become bounded (refer also to a comparison with an inertial scenario - Figure 7).

As for the filter estimation, the drift state, $\mathbf{d} = (d_x, d_y, d_z)^T$, (Figure 5) is fully estimated due to the ideal relative rotation measurement (however, as will be seen next, the precision of this measurement in reality is not sufficient for estimating drift). The bias state, $\mathbf{b} = (b_x, b_y, b_z)^T$, (Figure 6) is estimated in the $z$-direction.

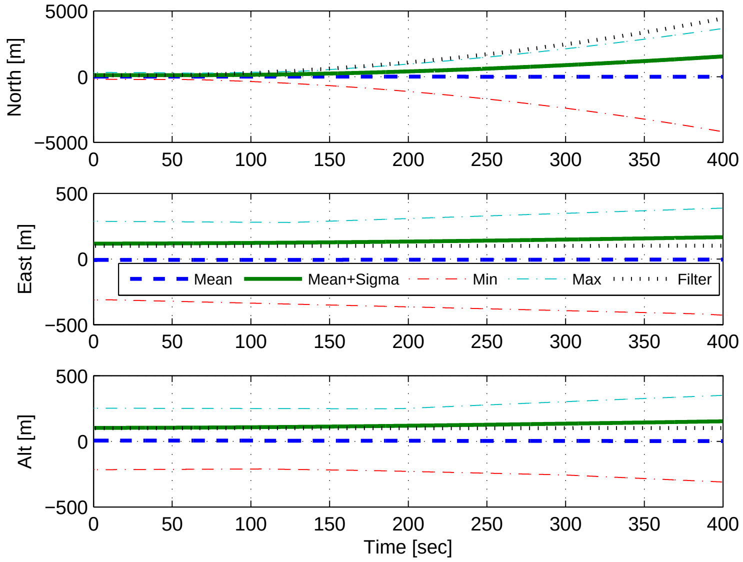


**Figure 2. Position errors statistics vs. filter covariance (Monte-Carlo run). Examined scenario: Straight and level north-heading flight; Ideal pseudo-measurements. Errors normal to the flight heading are reduced, errors along the flight heading are not diminished due to lack of observability.**

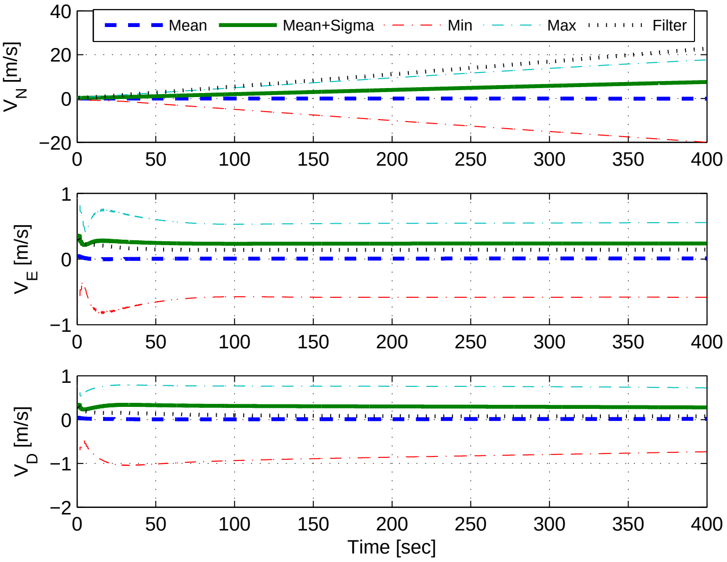


Figure 3. Velocity errors statistics vs. filter covariance (Monte-Carlo run). Examined scenario: Straight and level north-heading flight; Ideal pseudo-measurements. Errors normal to the flight heading are reduced and blocked, but not nullified due to errors associated with the DCM transforming the translation measurement from the camera system to the LLLN system. Errors along the flight heading are not diminished due to lack of observability.

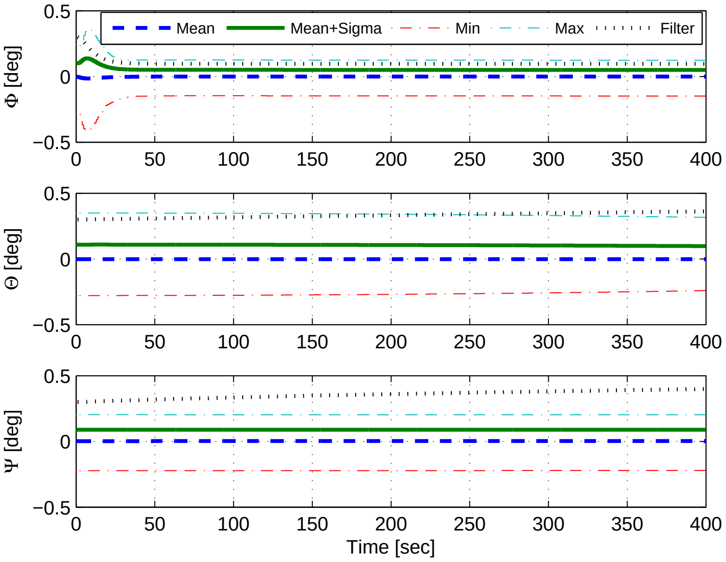


Figure 4. Euler angle errors statistics vs. filter covariance (Monte-Carlo run). Examined scenario: Straight and level north-heading flight; Ideal pseudo-measurements. Roll angle error, $\Delta\Phi$, is significantly reduced and bounded; pitch and yaw angles errors $(\Delta\Theta, \Delta\Psi)$, development is arrested. See Figure 7 for a comparison with an inertial scenario.

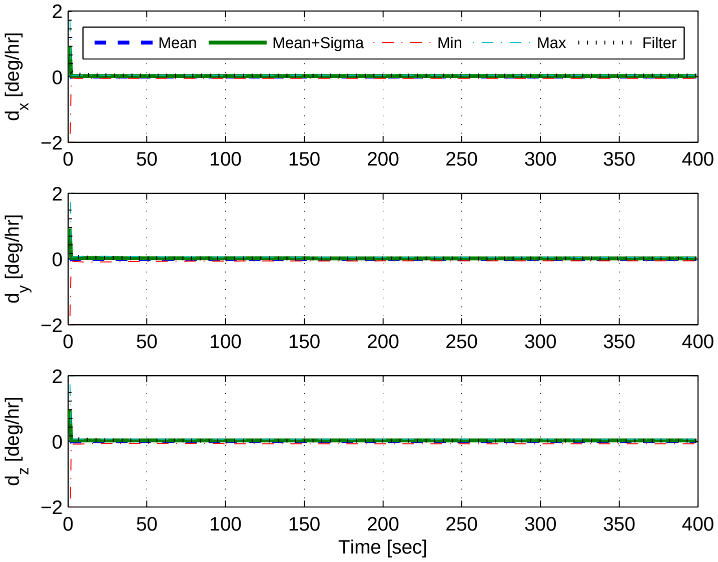


**Figure 5. Drift estimation errors statistics vs. filter covariance (Monte-Carlo run). Examined scenario: Straight and level north-heading flight; Ideal pseudo-measurements. Full drift estimation due to ideal relative rotation measurement.**

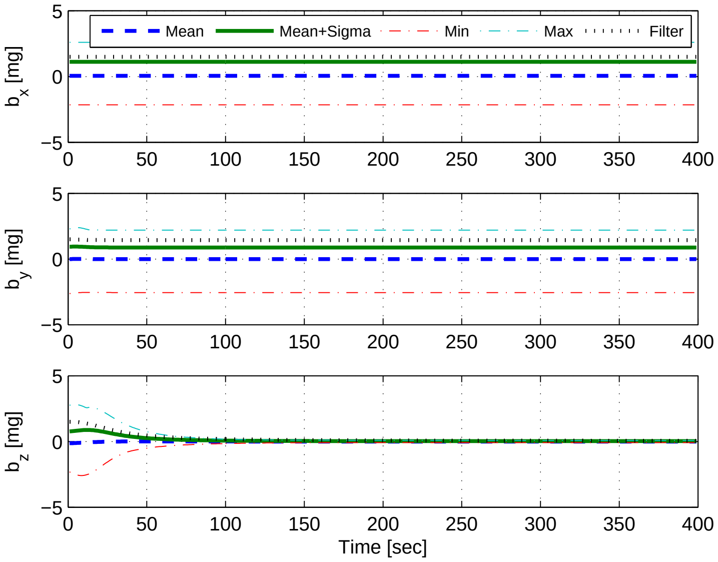


**Figure 6. Bias estimation errors statistics vs. filter covariance (Monte-Carlo run). Examined scenario: Straight and level north-heading flight; Ideal pseudo-measurements. The bias in the $z$ direction is estimated after about 50 sec.**

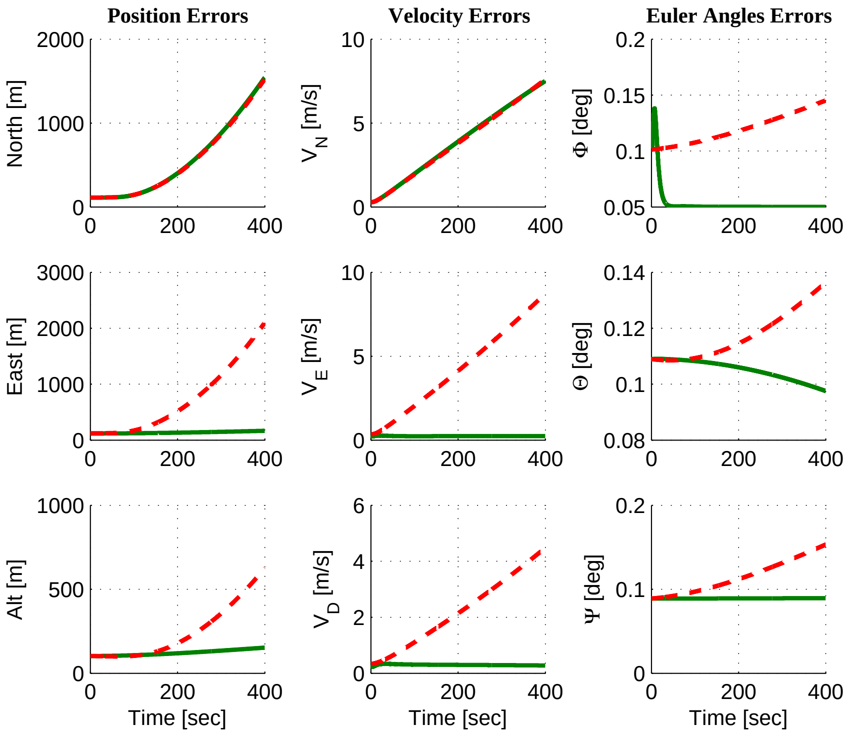


**Figure 7. Comparison to an inertial scenario: Solid line - navigation errors ($1\sigma$) with ideal pseudo-measurements; Dashed line - navigation errors ($1\sigma$) in an inertial scenario.**

## B. Two-View Aided Navigation: Experimental Results Based on Real Images Taken by a Wide-FOV Camera

This section presents experimental results of vision-aided navigation based on the two-view motion estimation method. A wide-FOV camera and high-texture scenes are used, establishing a valid scenario for fundamental matrix estimation (cf. Section III-B.1 in Ref. 1). The precision of motion parameters extracted from the estimated fundamental matrix is given in Figure 7 in Ref. 1. In the following discussion, results are presented for two different platform trajectories: straight and level trajectory; and a trajectory with a maneuver.

### *1. Straight and Level Trajectory*

The trajectory in this experiment consists of a straight and level north-heading flight for 400 sec at 1600 m above sea level (height above ground ranges from 600 to 1300 m) and a velocity of 150 m/s. The image-based relative motion measurements were injected at a 1 Hz frequency. Hence, 400 images were captured from Google Earth. The same IMU errors and initial navigation errors as in Table 1 were assumed.

Figures 8-11 provide the experimental results. The following three scenarios are shown:

- Inertial scenario.
- Real-images-based measurements; *without* a fictitious velocity measurement.
- Real-images-based measurements; *with* a fictitious velocity measurement.

It can be seen that the errors in position and velocity normal to flight heading (Figures 8 and 9) are significantly reduced compared to the inertial scenario, regardless of whether the fictitious velocity measurement was applied or not. As expected (cf. Section IV-A), the velocity errors are not nullified since the translation measurements are not ideal. The errors along the flight heading (north) behave as in the inertial scenario when the fictitious velocity measurement is applied, and are much degraded when this measurement is not applied.

The errors in the Euler angles are shown in Figure 10. The roll angle error $\Delta\Phi$ is nicely estimated and is nearly nullified (after 100 sec of flight in the current filter tuning); The pitch angle error $\Delta\Theta$ coincides with the error in an inertial scenario in case the fictitious velocity measurement is applied, and is enlarged otherwise (in accordance with the velocity error in the north direction). The relative rotation measurement precision does not allow estimation of the drift, since drift contribution to the angular motion in 1 sec (which is the time between two measurements) is 0.00027 deg (for a 1 deg/hr drift) while the measurement precision is only around 0.2 deg. As a consequence, the drift state is not estimated (Figure 11) and the errors in pitch (assuming fictitious velocity measurement is applied) and yaw angles are not restrained compared to an inertial scenario (in contrast to the situation when ideal pseudo-measurements are used).

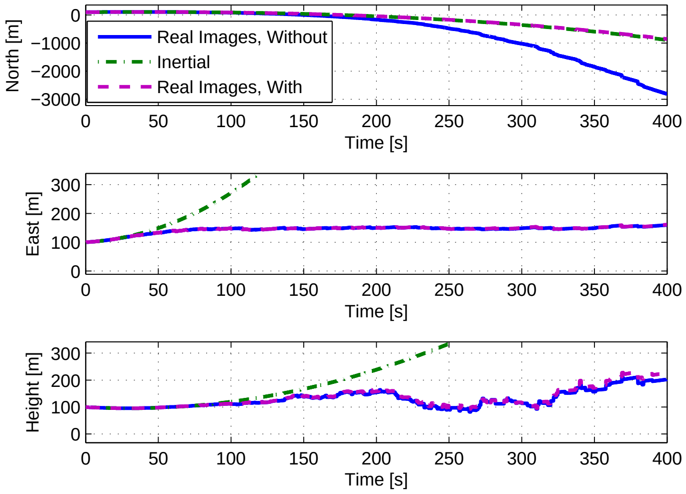


**Figure 8. Position errors. Real-images-based relative motion measurements, straight and level north-heading trajectory. Three curves are presented: inertial scenario, fusion of real imagery measurements with and without the fictitious velocity measurement. Errors normal to the flight heading are considerably reduced compared to the inertial scenario; inertial behavior of the errors along the flight heading is obtained if fictitious velocity measurement is applied. Significantly larger errors along the flight heading are obtained if the fictitious measurement is not applied.**

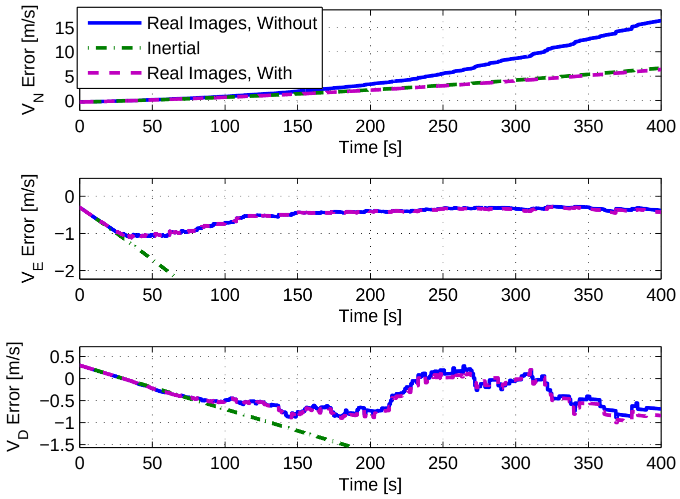


**Figure 9.** Velocity errors. Real-images-based relative motion measurements, straight and level north-heading trajectory. Three curves are presented: inertial scenario, fusion of real imagery measurements with and without the fictitious velocity measurement. Errors normal to the flight heading are considerably reduced compared to the inertial scenario, though not nullified due to 1) errors in the DCM transforming the translation measurement from camera the system to the LLLN system, and 2) imperfect translation measurements. Inertial behavior of errors along the flight heading is obtained if the fictitious velocity measurement is applied. Significantly larger errors along the flight heading are obtained if the fictitious measurement is not applied.

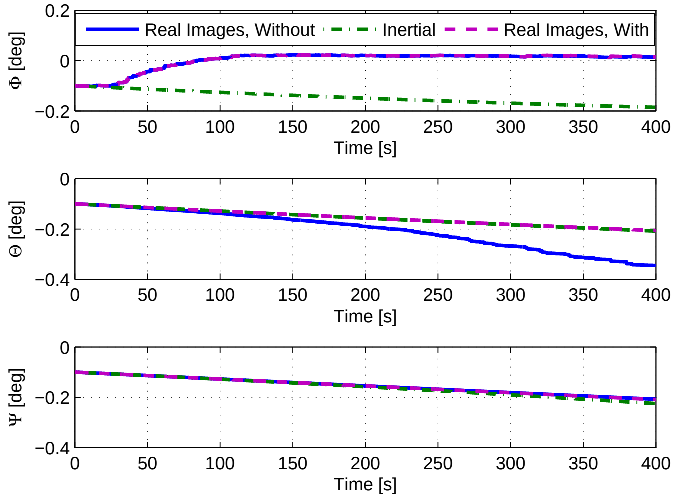


**Figure 10.** Euler angle errors. Real images based relative motion measurements, straight and level north-heading trajectory. Three curves are presented: inertial scenario, fusion of real imagery based measurements with and without the velocity fictitious measurement. Roll angle error is nearly nullified. Inertial pitch angle error in case the fictitious velocity measurement is applied, and enlarged error otherwise (due to enlarged velocity errors along the flight heading).

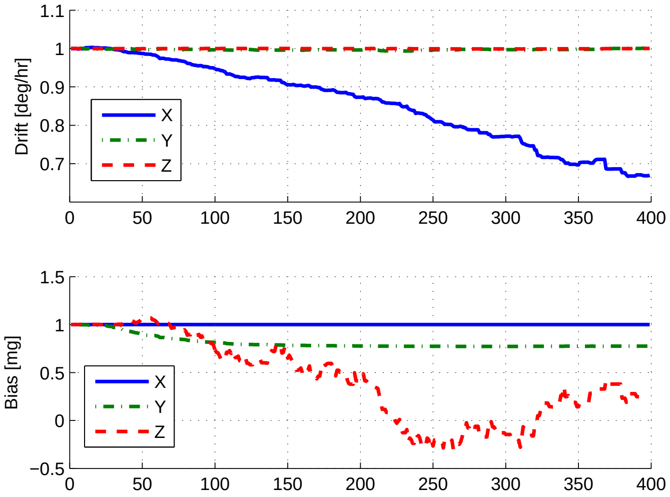


**Figure 11. Filter estimations errors. Real-images-based relative motion measurements with fictitious velocity measurements, straight and level north-heading trajectory. The drift state is not estimated due to insufficient precision of the relative rotation measurement. Slow estimation of the $z$-bias.**

### *2. Trajectory With a Maneuver*

The platform trajectory in this experiment consists of the following three segments:

- Straight and level north-heading flight ($0 \leq t < 70$ sec).
- 90-degree turn west with a constant 15 degrees roll angle (70 sec $\leq t <$ 160 sec).
- Straight and level west-heading flight (160 sec $\leq t \leq$ 210 sec).

The velocity and height above sea level are identical to those in the preceding section. The trajectory is described in Figure 12.

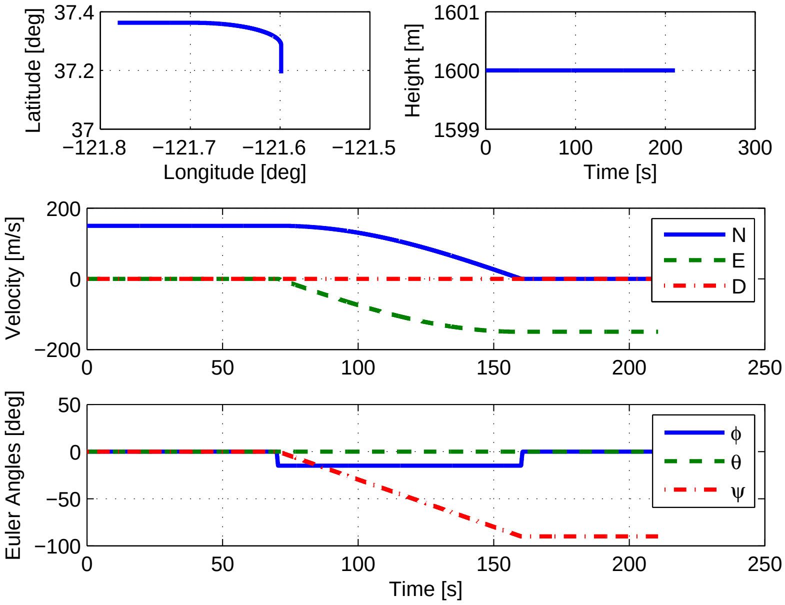


**Figure 12. Description of a trajectory with a maneuver.**

Figures 13-16 present the experimental results. Each figure contains the following three curves: Navigation errors fusing relative measurements based on real images, square root of the filter covariance and navigation errors in an inertial scenario for comparison.

During the first trajectory segment (north-heading straight and level flight, $0 \leq t <$ 70 sec), position and velocity errors (Figure 13 and 14) in the east and down directions are considerably reduced, while errors in the north direction coincide with an inertial scenario due to lack of observability, as is the case in the first trajectory. Roll angle error $\Delta\Phi$ (Figure 15) is also reduced during this segment.

The second trajectory segment comprises a 90 degrees turn with a constant roll angle $\Phi$ (70 sec $\leq t <$ 160 sec); the motion dynamics of this maneuver excite variables that were nullified during the straight and level flight (e. g. variables of the accelerometer measurements,

$\mathbf{f}_{NED}$) leading to a partial estimation of the pitch angle error (Figure 15). Furthermore, the north axis gradually becomes observable on account of the east axis, which leads to reduction in the north velocity and position errors.

During the last trajectory segment (west-heading straight and level flight, 160 sec $\leq t <$ 210 sec), the unobservable direction is east and thus position and velocity errors in this direction grow inertially, while errors in north and down directions are reduced.

As in the straight and level trajectory, the precision of relative rotation measurements is not sufficient for drift estimation (the rotation measurements were rejected by the filter).

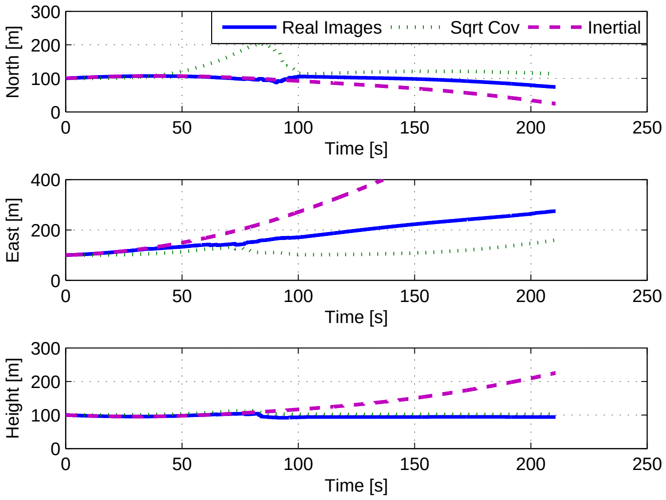


**Figure 13. Position errors. Real-images-based relative motion measurements, trajectory with a maneuver. Reduced position errors compared to an inertial scenario.**

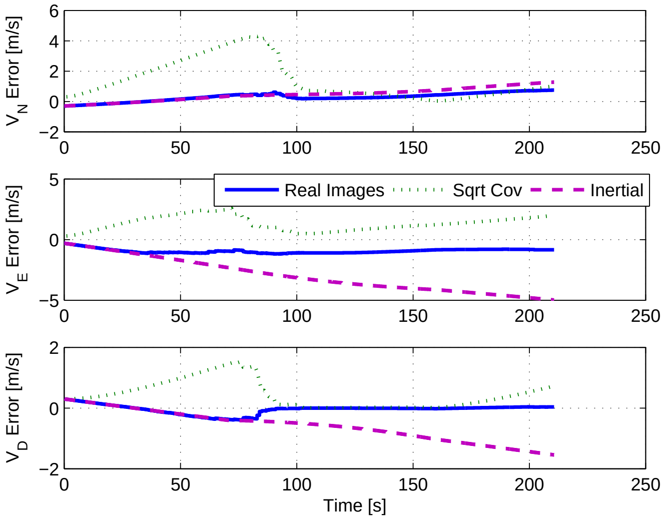


**Figure 14. Velocity errors. Real-images-based relative motion measurements, trajectory with a maneuver. The velocity errors are significantly reduced compared to the inertial scenario: For $0 \le t < 70$ sec, the platform performs a straight and level north-heading flight, therefore the errors are reduced in the east and down axes; during the maneuver phase, $70 \text{ sec} \le t < 160$ sec, the north direction becomes gradually observable on account of the east direction and thus the errors in the north direction are partially diminished; when the flight heading is west at $160 \text{ sec} \le t < 210$ sec, the east velocity errors are enlarged as this direction is unobservable, while north and down errors are reduced.**

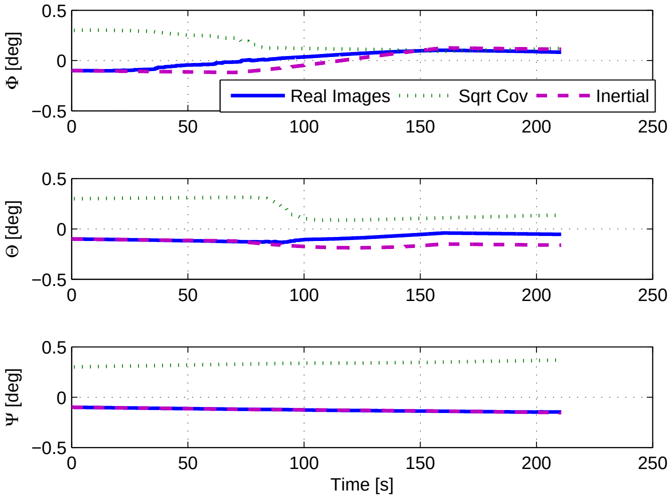


**Figure 15. Euler angle errors. Real-images-based relative motion measurements, trajectory with a maneuver. Roll angle error is well estimated during the north-heading trajectory segment ($0 \le t < 70$ sec); pitch angle error is partially reduced during the maneuver phase ($70 \text{ sec} \le t < 160$ sec). Yaw angle error inertial development due to lack of observability.**

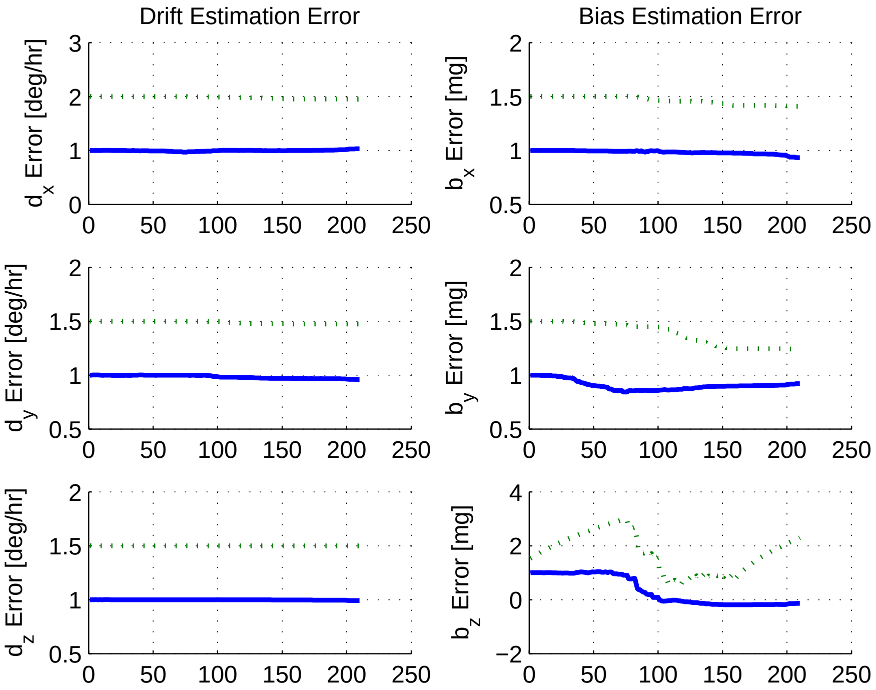


**Figure 16. Filter estimations. Real-images-based relative motion measurements, trajectory with a maneuver. Solid line - estimation error; Dotted line - filter covariance. Drift is not estimated due to insufficient relative rotation measurement precision. The $z$-bias is estimated as well as partial $y$-bias estimation.**

## C. Mosaic vs. Two-View Aided Navigation: Experimental Results Based on Real Images Taken by a Narrow-FOV Camera

This section demonstrates the superior performance of mosaic-aided navigation over two-view motion estimation. The examined scenario consists of a narrow-FOV camera ($5^o \times 3^o$) and a low-texture scene. The platform performs a straight and level north-heading flight.

The experiment consisted of 50 sec of inertial flight, followed by a 50 sec of vision-aided phase, during which the mosaic- and two-view-based motion estimations were injected into the navigation system. The last phase is another inertial navigation flight segment for 50 sec. Figures 17 - 20 provide the experimental results. Each figure compares the navigation performance for the two examined methods (mosaic and two-view). In addition, INS performance is given for reference.

The enhanced performance of the mosaic-based aiding can be clearly seen in Figures 17 and 18: During the vision-aided phase, the position and velocity errors perpendicular to the flight heading are significantly reduced. The mosaic-based aiding yields better results due to more accurate vision-based motion estimation. It can be concluded from these graphs that the number of measurements accepted by the filter is considerably higher in case of the mosaic framework (between 60 sec and 80 sec, all the measurements in the two-view method were rejected by the filter). As for the roll angle error (Figure 19), although this error is smaller with the two-view method, it is expected to reach higher values if more measurements were accepted by the filter.

When examining the behavior of the navigation errors in the inertial segment (after the vision-aided phase), one can notice the slow development of inertial errors when using mosaic aiding. The reason for this is the improved bias estimation compared to the estimation using the two-view method, as shown in Figure 20: $b_z$ is almost exactly estimated and thus it does not contribute to the growth of inertial position and velocity errors in the down axis.

It should be noted that the drift state was not estimated at all, since all the relative rotation measurements were rejected by the filter due to their low quality. This is not surprising, since these measurements were not accepted even in the much easier scenario of a wide-FOV camera (cf. Section IV-B).

The relative motion measurements have another interesting effect: Although the position error does not have a representation in the translation measurement matrix (cf. Eq. (12)), the measurements still reduce the position errors (Figure 17), due to the developing cross-correlation terms in the covariance matrix of the state vector.

Figure 21 compares the filter covariance to the actual developed errors. As seen, the covariance is consistent overall. Better tuning might have yielded improved results. For example, in the last segment of the inertial flight (after t=100 sec), the covariance development rate does not match the actual rate of the developing inertial navigation errors. After the vision-aided segment, part of the IMU error parameters are estimated by the filter (e. g. $b_z$) and are used to correct the actual IMU measurements. As a consequence, the actual IMU measurements injected into the navigation system are corrupted by the residual IMU parameters only, resulting in a much slower development of navigation errors. One possible alternative to account for this behavior is to perform a dynamic adjustment of the filter noise covariance matrix $Q$ as a function of the actual covariance values of the estimated IMU states.

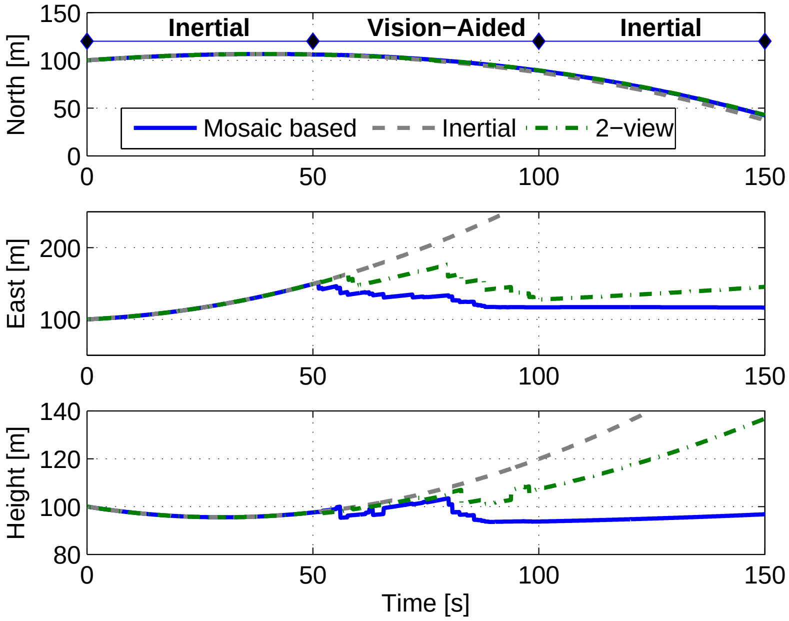


**Figure 17. Position errors. Vision-aided navigation: Mosaic aiding vs. two-view framework. Inertial error development in the north direction due to lack of observability. Reduced errors in the east and down directions, with a significant improvement in favor of the mosaic aiding.**

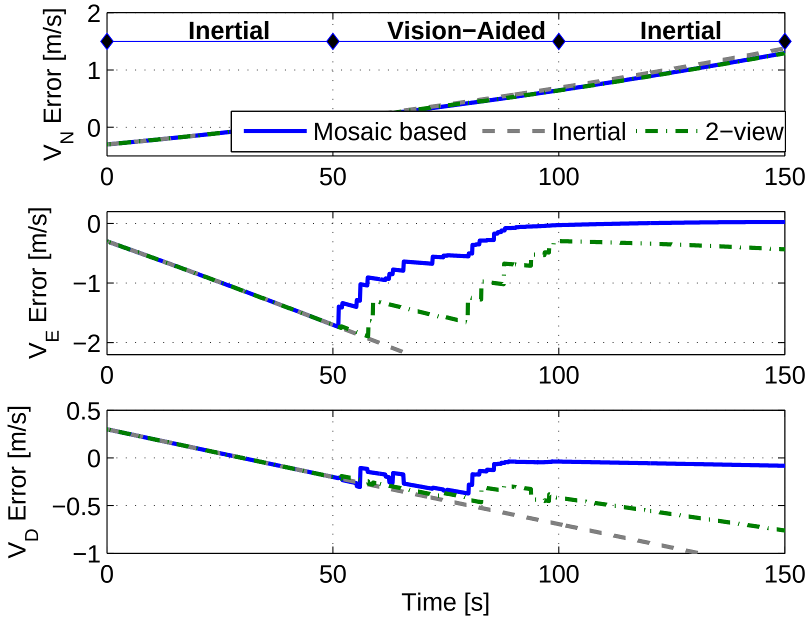


Figure 18. Velocity errors. Vision-aided navigation: Mosaic aiding vs. two-view aiding. Inertial error development in the north direction due to lack of observability. Reduced errors in the east and down directions, with a significant improvement in favor of the mosaic aiding.

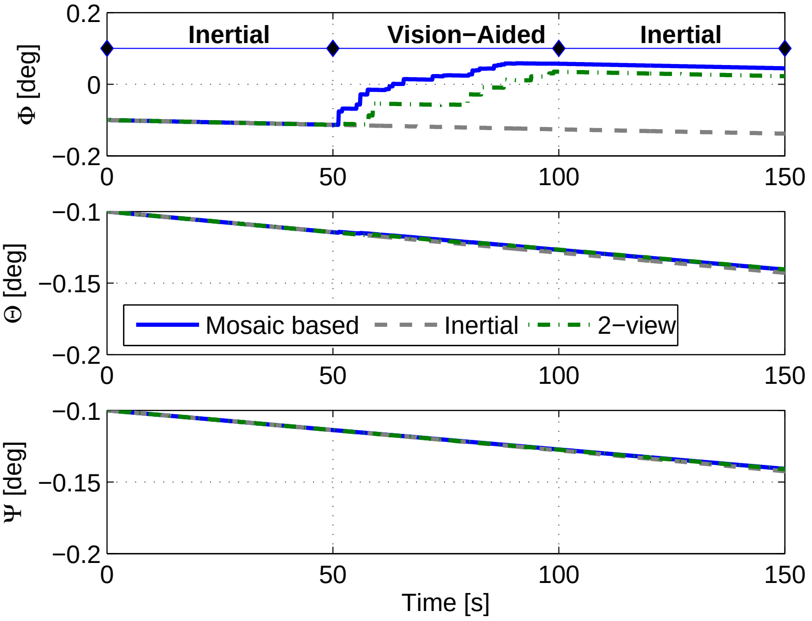


Figure 19. Euler angle errors. Vision-aided navigation: Mosaic aiding vs. two-view aiding. Roll angle error estimation for both motion estimation methods. Pitch and yaw angles errors are not reduced due to lack of observability.

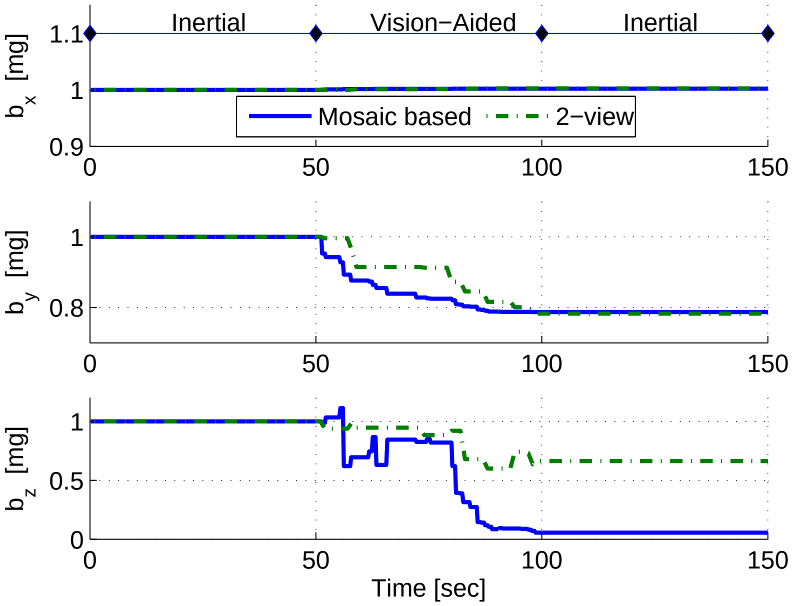
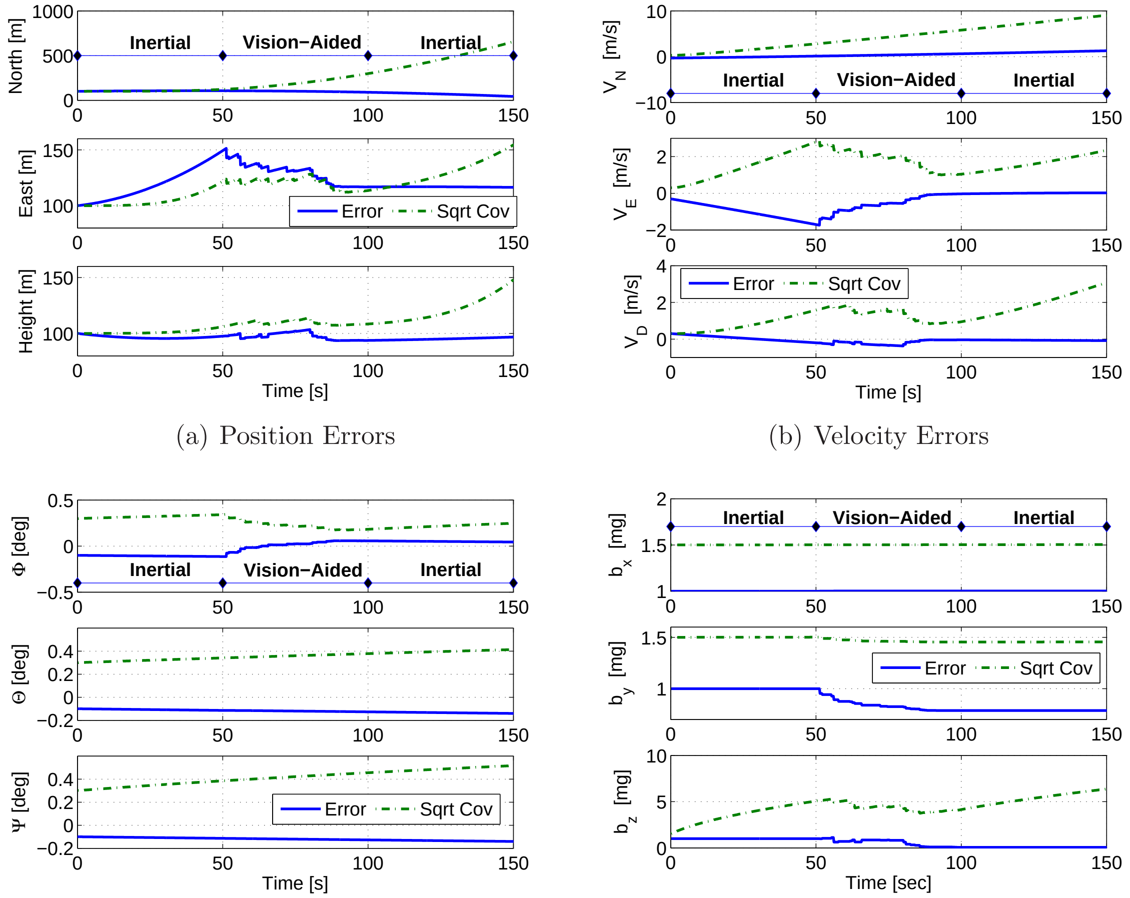


**Figure 20. Bias estimation errors. Vision-aided navigation: Mosaic aiding vs. two-view aiding. Considerably improved $b_z$ estimation in favor of the mosaic aiding.**



(a) Position Errors

(b) Velocity Errors

(c) Euler Angles Errors

(d) Bias Estimation Errors

**Figure 21. Actual navigation errors vs. filter covariance - Mosaic aiding. A consistent overall behavior of filter covariances, compared to the actual errors.**

# V. Conclusions

This paper presented a method for fusing mosaic-based motion estimation with a standard inertial navigation system of an airborne platform. This motion estimation was calculated by utilizing the on-line mosaic construction process and camera scanning procedure, as described in Part I of this work.[1]

A measurement model was formulated for fusing mosaic-based motion estimation with inertial navigation data using a Kalman filter. An extensive performance evaluation of the developed mosaic-aided navigation method was performed. The performance evaluation consisted of a statistical simulation study assuming ideal pseudo-measurements, and experiments involving realistic scenarios based on real imagery from Google Earth. These experiments included implementation of camera scanning and mosaic construction.

A comparison between the proposed method for vision-aided navigation using mosaic-based motion estimation and the same method utilizing the standard two-view motion estimation was carried out, focusing on difficult scenarios that involved a narrow-field-of-view camera and low-texture scenes. This comparison showed considerable improvement in navigation errors when mosaic-based aiding is used. The performance study indicated that the position and velocity errors normal to the flight heading, as well as the roll angle, can be significantly reduced, despite the imperfect nature of the estimated motion parameters. Errors along the flight heading were not reduced due to lack of observability.

# Appendix A: Development of the Translation Measurement Matrix

This appendix presents the development of the translation measurement matrix. We start with Eq. (11):

$$\left[T_{L_2}^{L_1}\left[-\frac{1}{6}A_s(t_1)T_{L_1}^{B_1}\mathbf{d}\cdot(\Delta t)^3+\frac{1}{2}\left[A_s(t_1)\Delta\mathbf{\Psi}(t_1)+T_{L_1}^{B_1}\mathbf{b}\right](\Delta t)^2+\Delta\mathbf{V}(t_1)\Delta t\right]\right]^{\wedge}\hat{\mathbf{t}}_{1\to 2}^{L_2}$$
$$+\mathbf{v}=\mathbf{z}_{translation} \quad (44)$$

¿From Eq. (44), one can conclude that the translation measurement equation is of the following form:

$$\mathbf{z}_{translation}=H^{tr}\mathbf{X}+\mathbf{v} \quad (45)$$

where

$$H^{tr}=\begin{bmatrix}0_{3\times 3} & H_{\Delta V}^{tr} & H_{\Delta\Psi}^{tr} & H_d^{tr} & H_b^{tr}\end{bmatrix} \quad (46)$$

and $\mathbf{X}$ is defined in Eq. (3).

Next, we elaborate on each submatrix of $H^{tr}$:

1. Velocity errors

$$\left[T_{L_2}^{L_1}\Delta\mathbf{V}(t_1)\Delta t\right]^{\wedge}\hat{\mathbf{t}}_{1\to 2}^{L_2} = -\left[\hat{\mathbf{t}}_{1\to 2}^{L_2}\right]^{\wedge} T_{L_2}^{L_1}\Delta\mathbf{V}(t_1)\Delta t$$

$$\Rightarrow H_{\Delta V}^{tr} = -\left[\hat{\mathbf{t}}_{1\to 2}^{L_2}\right]^{\wedge} T_{L_2}^{L_1}\Delta t \tag{47}$$

2. Attitude errors

$$\left[\tfrac{1}{2}T_{L_2}^{L_1}A_s(t_1)\Delta\mathbf{\Psi}(t_1)(\Delta t)^2\right]^{\wedge}\hat{\mathbf{t}}_{1\to 2}^{L_2} = -\tfrac{1}{2}\left[\hat{\mathbf{t}}_{1\to 2}^{L_2}\right]^{\wedge} T_{L_2}^{L_1}A_s(t_1)\Delta\mathbf{\Psi}(t_1)(\Delta t)^2$$

$$\Rightarrow H_{\Delta\Psi}^{tr} = -\tfrac{1}{2}\left[\hat{\mathbf{t}}_{1\to 2}^{L_2}\right]^{\wedge} T_{L_2}^{L_1}A_s(t_1)(\Delta t)^2 \tag{48}$$

3. Drift

$$-\tfrac{1}{6}\left[T_{L_2}^{L_1}A_s(t_1)T_{L_1}^{B_1}\mathbf{d}(\Delta t)^3\right]^{\wedge}\hat{\mathbf{t}}_{1\to 2}^{L_2} = \tfrac{1}{6}\left[\hat{\mathbf{t}}_{1\to 2}^{L_2}\right]^{\wedge} T_{L_2}^{L_1}A_s(t_1)T_{L_1}^{B_1}\mathbf{d}(\Delta t)^3$$

$$\Rightarrow H_{d}^{tr} = \tfrac{1}{6}\left[\hat{\mathbf{t}}_{1\to 2}^{L_2}\right]^{\wedge} T_{L_2}^{L_1}A_s(t_1)T_{L_1}^{B_1}(\Delta t)^3 \tag{49}$$

4. Bias

$$\tfrac{1}{2}\left[T_{L_2}^{L_1}T_{L_1}^{B_1}\mathbf{b}(\Delta t)^2\right]^{\wedge}\hat{\mathbf{t}}_{1\to 2}^{L_2} = -\tfrac{1}{2}\left[\hat{\mathbf{t}}_{1\to 2}^{L_2}\right]^{\wedge} T_{L_2}^{L_1}T_{L_1}^{B_1}\mathbf{b}(\Delta t)^2$$

$$\Rightarrow H_{b}^{tr} = -\tfrac{1}{2}\left[\hat{\mathbf{t}}_{1\to 2}^{L_2}\right]^{\wedge} T_{L_2}^{L_1}T_{L_1}^{B_1}(\Delta t)^2 \tag{50}$$

## Appendix B: A Model for Inertial Navigation Errors

The development of inertial navigation errors can be expressed for short time instances using a simplified model of the state-space formulation. Eq. (51) presents such a model, in which the gyro and accelerometer errors are considered to be random constants and thus the state vector consists of the navigation error parameters (position error $\Delta\mathbf{P}$, velocity error $\Delta\mathbf{V}$ and Euler angles error $\Delta\mathbf{\Psi}$) and of an IMU error parametrization: The drift vector, $\mathbf{d}$, and the bias vector, $\mathbf{b}$. Note that the navigation errors ($\Delta\mathbf{P}$, $\Delta\mathbf{V}$ and $\Delta\mathbf{\Psi}$) are expressed in the LLLN system, while the states $\mathbf{d}_B$ and $\mathbf{b}_B$ are expressed in the body system. The residual IMU errors are considered to be white noise signals:

$$\dot{\mathbf{X}} = \Phi_c \mathbf{X} + \mathbf{w} \tag{51}$$

where $\mathbf{X}$ is defined in Eq. (3) and

$$\Phi_c = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & A_s & 0_{3\times3} & T_L^B \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -T_L^B & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} 0_{3\times1} \\ \mathbf{e_V} \\ \mathbf{e_\Psi} \\ 0_{3\times1} \\ 0_{3\times1} \end{bmatrix} \tag{52}$$

Here $T_L^B = T_{L_C}^B$ is the DCM transforming from the body system to the computed LLLN system and $A_s$ is the skew-symmetric matrix of the specific force vector $\mathbf{f}$ measured by the accelerometers:

$$A_s = \begin{bmatrix} 0 & -f_D & f_E \\ f_D & 0 & -f_N \\ -f_E & f_N & 0 \end{bmatrix} \tag{53}$$

Note that the first three entries of the state vector, $\mathbf{X}$, are represented in the LLLN system, whereas the last two entries are represented in the body system.

From Eq. (51), one can get the following explicit expressions for the navigation errors (these expressions are valid for short time intervals, $(t - t_0)$, or for a straight and level flight trajectories):

**Euler angles error:**

$$\Delta\mathbf{\Psi}(t) = -T_L^B \mathbf{d_B} \Delta t + \Delta\mathbf{\Psi}(t_0) \tag{54}$$

**Velocity error:**

$$\Delta\mathbf{V}(t) \;=\; -\frac{1}{2} A_s T_L^B \mathbf{d_B} (\Delta t)^2 + \left[A_s \Delta\mathbf{\Psi}(t_0) + T_L^B \mathbf{b_B}\right] \Delta t + \Delta\mathbf{V}(t_0) \tag{55}$$

**Position error:**

$$\Delta\mathbf{P}(t) = -\frac{1}{6} A_s T_L^B \mathbf{d_B} (\Delta t)^3 + \frac{1}{2}\left[A_s \Delta\mathbf{\Psi}(t_0) + T_L^B \mathbf{b_B}\right] (\Delta t)^2 + \Delta\mathbf{V}(t_0)\Delta t + \Delta\mathbf{P}(t_0) \tag{56}$$

## References

[1] Indelman, V., Gurfil, P., Rivlin, E. and Rotstein,H., *Real-Time Mosaic-Aided Aircraft Navigation: I. Motion Estimation*, AIAA GN&C Conference, USA, 2009.

[2] Indelman, V., Gurfil, P., Rivlin, E. and Rotstein,H., *Navigation Performance Enhancement Using Rotation and Translation Measurements from Online Mosaicking*, AIAA GN&C Conference, SC, USA, 2007.

[3] Indelman, V., Gurfil, P., Rivlin, E. and Rotstein,H., *Navigation Aiding Using On-Line Mosaicking*, IEEE/ION PLANS, California, USA, 2008.

[4] Merhav, S. and Bresler, Y., "On-Line Vehicle Motion Estimation from Visual Terrain Information Part 1: Recursive Image Registration," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 22, No. 5, 1986, pp. 583–587.

[5] Gurfil, P. and Rotstein, H., "Partial Aircraft State Estimation from Visual Motion Using the Subspace Constraints Approach," *Journal of Guidance, Control and Dynamics*, Vol. 24, No. 5, July 2001, pp. 1016–1028.

[6] Soatto, S. and Perona, P., "Recursive 3-D Visual Motion Estimation Using Subspace Constraints," *International Journal of Computer Vision*, Vol. 22, No. 3, 1997, pp. 235–259.

[7] Oshman, Y. and Menis, B., "Maximum a Posteriori Image Registration/Motion Estimation," *Journal of Guidance, Control and Dynamics*, Vol. 17, No. 5, September-October 1994, pp. 1115–1123.

[8] Diel, D., DeBitetto, P. and Teller, S., "Epipolar Constraints for Vision-Aided Inertial Navigation," *Proceedings of the IEEE Workshop on Motion and Video Computing*, Vol. 2, January 2005, pp. 221–228.

[9] Roumeliotis, S., Johnson, A. and Montgomery, J., "Augmenting Inertial Navigation with Image-Based Motion Estimation," *IEEE International Conference on Robotics and Automation*, Vol. 4, 2002, pp. 4326–4333.

[10] Merhav, S. and Bresler, Y., "On-Line Vehicle Motion Estimation from Visual Terrain Information Part 2: Ground Velocity and Position Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 22, No. 5, 1986, pp. 588–604.

[11] Lerner, R., Rivlin, E., and Rotstein, H., "Pose and Motion Recovery from Feature Correspondences and a Digital Terrain Map," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 9, September 2006, pp. 1404–1417.

[12] Sim, D., Park, R., Kim, R., Lee, S., and Kim, I., "Integrated Position Estimation Using Aerial Image Sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 1, January 2002, pp. 1–18.

[13] Gracias, N., Zwaan, S., Bernardino, A. and Santos-Victor, J., "Results on Underwater Mosaic-based Navigation," *IEEE OCEANS*, Vol. 3, 2002, pp. 1588–1594.

[14] Gracias, N. and Santos-Victor, J., "Underwater Video Mosaics as Visual Navigation Maps," *Computer Vision and Image Understanding*, Vol. 79, 2000, pp. 66–91.

[15] Davison, A.J., Reid, I.D. and Molton, N.D., "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, 2007.