

Research Article

Vision-Based Unmanned Aerial Vehicle Navigation Using Geo-Referenced Information

Gianpaolo Conte and Patrick Doherty

Artificial Intelligence and Integrated Computer System Division, Department of Computer and Information Science, Linköping University, 58183 Linköping, Sweden

Correspondence should be addressed to Gianpaolo Conte, giaco@ida.liu.se

Received 31 July 2008; Revised 27 November 2008; Accepted 23 April 2009

Recommended by Matthijs Spaan

This paper investigates the possibility of augmenting an Unmanned Aerial Vehicle (UAV) navigation system with a passive video camera in order to cope with long-term GPS outages. The paper proposes a vision-based navigation architecture which combines inertial sensors, visual odometry, and registration of the on-board video to a geo-referenced aerial image. The vision-aided navigation system developed is capable of providing high-rate and drift-free state estimation for UAV autonomous navigation without the GPS system. Due to the use of image-to-map registration for absolute position calculation, drift-free position performance depends on the structural characteristics of the terrain. Experimental evaluation of the approach based on offline flight data is provided. In addition the architecture proposed has been implemented on-board an experimental UAV helicopter platform and tested during vision-based autonomous flights.

Copyright © 2009 G. Conte and P. Doherty. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

One of the main concerns which prevents the use of UAV systems in populated areas is the safety issue. State-of-the-art UAVs are still not able to guarantee an acceptable level of safety to convince aviation authorities to authorize the use of such a system in populated areas (except in rare cases such as war zones). There are several problems which have to be solved before unmanned aircraft can be introduced into civilian airspace. One of them is GPS vulnerability [1].

Autonomous unmanned aerial vehicles usually rely on a GPS position signal which, combined with inertial measurement unit (IMU) data, provide high-rate and drift-free state estimation suitable for control purposes. Small UAVs are usually equipped with low-performance IMUs due to their limited payload capabilities. In such platforms, the loss of GPS signal even for a few seconds can be catastrophic due to the high drift rate of the IMU installed on-board. The GPS signal becomes unreliable when operating close to obstacles due to multipath reflections. In addition, jamming of GPS has arisen as a major concern for users due to the availability of GPS jamming technology on the market. Therefore UAVs

which rely blindly on a GPS signal are quite vulnerable to malicious actions. For this reason, this paper proposes a navigation system which can cope with GPS outages. The approach presented fuses information from inertial sensors with information from a vision system based on a passive monocular video camera. The vision system replaces the GPS signal combining position information from visual odometry and geo-referenced imagery. Geo-referenced satellite or aerial images must be available on-board UAV beforehand or downloaded in flight. The growing availability of high-resolution satellite images (e.g., provided by Google Earth) makes this topic very interesting and timely.

The vision-based architecture developed is depicted in Figure 2 and is composed of an error dynamics Kalman filter (KF) that estimates the navigation errors of the INS and a separate Bayesian filter named *point-mass filter* (PMF) [2] which estimates the absolute position of the UAV on the horizontal plane fusing together visual odometry and image registration information. The 2D position estimated from the PMF, together with barometric altitude information obtained from an on-board barometer, is used as position measurement to update the KF.



FIGURE 1: The Rmax helicopter.

Odometry and image registration are complementary position information sources. The KLT feature tracker [3] is used to track corner features in the on-board video image from subsequent frames. A homography-based odometry function uses the KLT results to calculate the distance traveled by the UAV. Since the distance calculated by the odometry is affected by drift, a mechanism which compensates for the drift error is still needed. For this purpose information from geo-referenced imagery is used.

The use of reference images for aircraft localization purposes is also investigated in [4]. A reference image matching method which makes use of the Hausdorff distance using contour images representation is explored. The work focuses mainly on the image processing issues and gives less emphasis to architectural and fusion schemes which are of focus in this paper.

An emerging technique to solve localization problems is Simultaneous Localization and Mapping (SLAM). The goal of SLAM is to localize a robot in the environment while mapping it at the same time. Prior knowledge of the environment is not required. In SLAM approaches, an internal representation of the world is built on-line in the form of a landmarks database. Such a representation is then used for localization purposes. For indoor robotic applications SLAM is already a standard localization technique. More challenging is the use of such technique in large outdoor environments. Some examples of SLAM applied to aerial vehicles can be found in [5–7].

There also exist other kinds of terrain navigation methods which are not based on aerial images but on terrain elevation models. A direct measurement of the flight altitude relative to the ground is required. Matching the ground elevation profile, measured with a radar altimeter for example, to an elevation database allows for aircraft localization. A commercial navigation system called TERrain NAVigation (TERNAV) is based on such a method. The navigation system has been implemented successfully on some military jet fighters and cruise missiles. In the case of small UAVs and more specifically for unmanned helicopters, this method does not appear to be appropriate. Compared to jet fighters, UAV helicopters cover short distances at very low speed, and so the altitude variation is quite poor in terms of allowing ground profile matching.

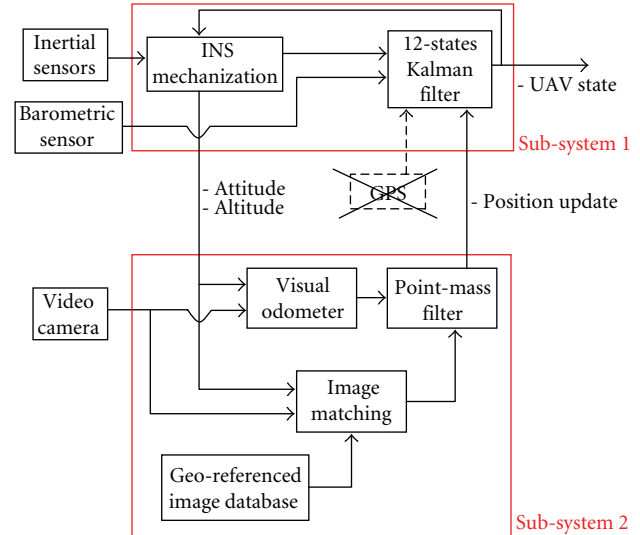


FIGURE 2: Sensor fusion architecture.

Advanced cruise missiles implement a complex navigation system based on GPS, TERNAV, and Digital Scene Matching Area Correlation (DSMAC). During the cruise phase the navigation is based on GPS and TERNAV. When the missile approaches the target, the DSMAC is used to increase the position accuracy. The DSMAC matches the on-board camera image to a reference image using correlation methods.

The terrain relative navigation problem is of great interest not only for terrestrial applications but also for future space missions. One of the requirements for the next lunar mission in which NASA/JPL is involved is to autonomously land within 100 meters of a predetermined location on the lunar surface. Traditional lunar landing approaches based on inertial sensing do not have the navigational precision to meet this requirement. A survey of the different terrain navigation approaches can be found in [8] where methods based on passive imaging and active range sensing are described.

The contribution of this work is to explore the possibility of using a single video camera to measure both relative displacement (odometry) and absolute position (image registration). We believe that this is a very practical and innovative concept. The sensor fusion architecture developed combines vision-based information together with inertial information in an original way resulting in a light weight system with real-time capabilities. The approach presented is implemented on a Yamaha Rmax unmanned helicopter and tested on-board during autonomous flight-test experiments.

2. Sensor Fusion Architecture

The sensor fusion architecture developed in this work is presented in Figure 2 and is composed of several modules. It can work in GPS modality or vision-based modality if the GPS is not available.

A traditional KF (subsystem 1) is used to fuse data from the inertial sensors (3 accelerometers and 3 gyros) with a position sensor (GPS or vision system in case the GPS is not available). An INS mechanization function performs the time integration of the inertial sensors while the KF function estimates the INS errors. The KF is implemented in the error dynamics form and uses 12 states: 3-dimensional position error, 3-dimensional velocity error, 3-attitude angle error, and 3 accelerometer biases. The error dynamics formulation of the KF is widely used in INS/GPS integration [9–11]. The estimated errors are used to correct the INS solution.

The vision system (subsystem 2) is responsible for computing the absolute UAV position on the horizontal plane. Such position is then used as a measurement update for the KF. The visual odometry computes the helicopter displacement by tracking a number of corner features in the image plane using the KLT algorithm [3]. The helicopter displacement is computed incrementally from consecutive image frames. Unfortunately the inherent errors in the computation of the features location accumulate when old features leave the frame and new ones are acquired, producing a drift in the UAV position. Such drift is less severe than the position drift derived from a pure integration of typical low-cost inertial sensors for small UAVs as it will be shown later in the paper. The image matching module provides drift-free position information which is integrated in the scheme through a grid-based Bayesian filtering approach. The matching between reference and on-board image is performed using the normalized cross-correlation algorithm [12].

The architecture components will be described in details in the following sections. Section 3 describes the homography-based visual odometry, Section 4 presents the image registration approach used, and Section 5 describes the sensor fusion algorithms.

3. Visual Odometry

Visual odometry for aerial navigation has been an object of great interest during the last decade [13–16]. The work in [13] has shown how a visual odometry is capable of stabilizing an autonomous helicopter in hovering conditions. In that work the odometry was based on a template matching technique where the matching between subsequent frames was obtained through sum of squared differences (SSDs) minimization of the gray scale value of the templates. The helicopter's attitude information was taken from an IMU sensor. Specialized hardware was expressly built for this experiment in order to properly synchronize attitude information with the video frame. Most of the recent work [14–16] on visual odometry for airborne applications is based on homography estimation under a planar scene assumption. In this case the relation between points of two images can be expressed as $x_2 \approx Hx_1$, where x_1 and x_2 are the corresponding points of two images 1 and 2 expressed in homogeneous coordinates, and H is the 3×3 homography matrix. The symbol \approx indicates that the relation is valid up to a scale factor. A point is expressed in homogeneous

coordinates when it is represented by equivalence classes of coordinate triples (kx, ky, k) , where k is a multiplicative factor. The camera rotation and displacement between two camera positions, c_1 and c_2 , can be computed from the homography matrix decomposition [17]:

$$H = K \left(R_{c_1}^{c_2} + \frac{1}{d} \vec{t}^{c_2} \vec{n}^{c_1 T} \right) K^{-1}, \quad (1)$$

where K is the camera calibration matrix determined with a camera calibration procedure, \vec{t}^{c_2} is the camera translation vector expressed in camera 2 reference system, $R_{c_1}^{c_2}$ is the rotation from camera 1 to camera 2, \vec{n}^{c_1} is the unit normal vector to the plane being observed and expressed in camera 1 reference system, and d is the distance of the principal point of camera 1 to the plane.

The visual odometry implemented in this work is based on robust homography estimation. The homography matrix H is estimated from a set of corresponding corner features being tracked from frame to frame. H can be estimated using direct linear transformation [17] with a minimum number of four feature points (the features must not be collinear). In practice the homography is estimated from a higher number of corresponding feature points (50 or more), and the random sample consensus (RANSAC) algorithm [18] is used to identify and then discard incorrect feature correspondences. The RANSAC is an efficient method to determine among the set of tracked features C the subset C_i of inliers features and subset C_o of outliers so that the estimate H is unaffected by wrong feature correspondences. Details on robust homography estimation using the RANSAC approach can be found in [17].

The feature tracker used in this work is based on the KLT algorithm. The algorithm selects a number of features in an image according to a “goodness” criteria described in [19]. Then it tries to reassociate the same features in the next image frame. The association is done by minimizing the sum of squared differences over patches taken around the features in the second image. Such association criteria gives good results when the feature displacement is not too large. Therefore it is important that the algorithm has a low execution time. The faster the algorithm, the more successful the association process.

In Figure 3 the RANSAC algorithm has been applied on a set of features tracked in two consecutive frames. In Figure 3(a) the feature displacements computed by the KLT algorithm are represented while in Figure 3(b) the set of outlier features has been detected and removed using RANSAC.

Once the homography matrix has been estimated, it can be decomposed into its rotation and translation components in the form of (1) using singular value decomposition as described in [17]. However, homography decomposition is a poorly conditioned problem especially when using cameras with a large focal length [20]. The problem arises also when the ratio between the flight altitude and the camera displacement is high. For this reason it is recommendable to use interframe rotation information from other sensor sources if available.

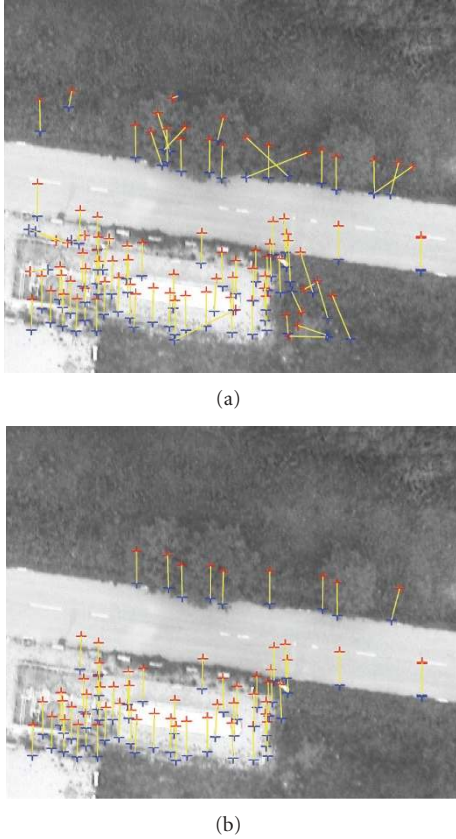


FIGURE 3: (a) displays the set of features tracked with the KLT algorithm. In (b) the outlier feature set has been identified and removed using the RANSAC algorithm.

The odometry presented in this work makes use of interframe rotation information coming from the KF. The solution presented in the remainder of this section makes use of the knowledge of the terrain slope; in other words the direction of the normal vector of the terrain is assumed to be known. In the experiment presented here the terrain will be considered nonsloped. Information about the terrain slope could be also extracted from a database if available.

The goal then is to compute the helicopter translation in the navigation reference system from (1). The coordinate transformation between the camera and the INS sensor is realized with a sequence of rotations. The translation between the two sensors will be neglected since the linear distance between them is small. The rotations sequence (2) aligns the navigation frame (n) to the camera frame (c) passing through intermediate reference systems named helicopter body frame (b) and camera gimbal frame (g) as follows:

$$R_c^n = R_b^g R_g^c, \quad (2)$$

where R_b^n is given by the KF, R_g^b is measured by the pan/tilt camera unit, and R_c^g is constant since the camera is rigidly mounted on the pan/tilt unit. A detailed description of the reference systems and rotation matrices adopted here can be found in [21]. The camera projection model used in

this work is a simple pin-hole model. The transformation between the image reference system and the camera reference system is realized through the calibration matrix K :

$$K = \begin{bmatrix} 0 & f_x & ox \\ -f_y & 0 & oy \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

where f_x , f_y are the camera focal lengths in the x and y directions, and $(ox \ oy)$ is the center point of the image in pixels. The camera's lens distortion compensation is not applied in this work. If $c1$ and $c2$ are two consecutive camera positions, then considering the following relationships:

$$\begin{aligned} R_{c1}^{c2} &= R_n^{c2} R_{c1}^n, \\ \vec{t}^{c2} &= R_n^{c2} \vec{t}^{nT}, \\ \vec{n}^{c1} &= R_n^{c1} \vec{n}^{nT} \end{aligned} \quad (4)$$

and substituting in (1) give

$$H = K R_n^{c2} \left(I + \frac{1}{d} \vec{t}^{nT} \vec{n}^{nT} \right) R_{c1}^n K^{-1}. \quad (5)$$

Considering that the rotation matrices are orthogonal (in such case the inverse coincides with the transposed), (5) can be rearranged as

$$\vec{t}^{nT} \vec{n}^{nT} = d \left(R_n^{c2T} K^{-1} H^* K R_{c1}^n - I \right), \quad (6)$$

where, in order to eliminate the scale ambiguity, the homography matrix has been normalized with the ninth element of H as follows:

$$H^* = \frac{H}{H_{3,3}}. \quad (7)$$

The distance to the ground d can be measured from an on-board laser altimeter. In case of flat ground, the differential barometric altitude, measured from an on-board barometric sensor between the take-off and the current flight position, can be used. Since in our environment the terrain is considered to be flat then $\vec{n}^n = [0, 0, 1]$. Indicating with RHS the right-hand side of (6), the north and east helicopter displacements computed by the odometry are

$$\begin{aligned} t_{\text{north}}^n &= \text{RHS}_{1,3}, \\ t_{\text{east}}^n &= \text{RHS}_{2,3}. \end{aligned} \quad (8)$$

To conclude this section, some considerations about the planar scene assumption will be made. Even if the terrain is not sloped, altitude variations between the ground level and roof top or tree top level are still present in the real world. The planar scene assumption is widely used for airborne application; however a simple calculation can give a rough idea of the error being introduced.

For a UAV in a level flight condition with the camera pointing perpendicularly downward, the 1D pin-hole camera projection model can be used to make a simple calculation:

$$\Delta x_p = f \frac{\Delta x_c}{d} \quad (9)$$

with f being the camera focal length, d the distance to the ground plane, Δx_p the pixel displacement in the camera image of an observed feature, and Δx_c the computed odometry camera displacement. If the observed feature is not on the ground plane but at a δd from the ground, the true camera displacement Δx_c^t can be expressed as a function of the erroneous camera displacement Δx_c as

$$\Delta x_c^t = \Delta x_c \left(1 - \frac{\delta d}{d} \right) \quad (10)$$

with $\epsilon = \delta d/d$ being the odometry error due to the depth variation. Then, if δd is the typical roof top height of an urban area, the higher the flight altitude d , the smaller the error ϵ is. Considering an equal number of features picked on the real ground plane and on the roof tops, the reference homography plane can be considered at average roof height over the ground, and the odometry error can be divided by 2. If a UAV is flying at an altitude of 150 meters over an urban area with a typical roof height at 15 meters, the odometry error derived from neglecting the height variation is about 5%.

4. Image Registration

Image registration is the process of overlaying two images of the same scene taken at different times, from different viewpoints and by different sensors. The registration geometrically aligns two images (the *reference* and *sensed* images). Image registration has been an active research field for many years, and it has a wide range of applications. A literature review on image registration can be found in [22, 23]. In this context it is used to extract global position information for terrain relative navigation.

Image registration is performed with a sequence of image transformations, including rotation, scaling, and translation which bring the sensed image to overlay precisely with the reference image. In this work, the reference and sensed images are aligned and scaled using the information provided by the KF. Once the images are aligned and scaled, the final match consists in finding a 2D translation which overlays the two images.

Two main approaches can be adopted to image registration: *correlation-based matching* and *pattern matching*. In the correlation-based matching the sensed image is placed at every pixel location in the reference image, and then, a similarity criteria is adopted to decide which location gives the best fit. In pattern matching approaches on the other hand, salient features (or landmarks) are detected in both images, and the registration is obtained by matching the set of features between the images. Both methods have advantages and disadvantages.

Methods based on correlation can be implemented very efficiently and are suitable for real-time applications. They can be applied also in areas with no distinct landmarks. However they are typically more sensitive to differences between the sensed and reference image than pattern matching approaches.

Methods based on pattern matching do not use image intensity values directly. The patterns are information on a higher level typically represented with geometrical models. This property makes such methods suitable for situations when the terrain presents distinct landmarks which are not affected by seasonal changes (i.e., roads, houses). If recognized, even a small landmark can make a large portion of terrain unique. This characteristic makes these methods quite dissimilar from correlation-based matching where small details in an image have low influence on the overall image similarity. On the other hand these methods work only if there are distinct landmarks in the terrain. In addition, a pattern detection algorithm is required before any matching method can be applied. A pattern matching approach which does not require geometrical models is the Scale Invariant Feature Transform (SIFT) method [24]. The reference and sensed images can be converted into feature vectors which can be compared for matching purposes. Knowledge about altitude and orientation of the camera relative to the terrain is not required for matching. Correlation methods are in general more efficient than SIFT because they do not require a search over image scale. In addition SIFT features do not have the capability to handle variation of illumination condition between reference and sensed images [8, 25].

This work makes use of a matching technique based on a correlation approach. The normalized cross-correlation of intensity images [12] is utilized. Before performing the cross-correlation, the sensed and reference images are scaled and aligned. The cross-correlation is the last step of the registration process, and it provides a measure of similarity between the two images.

The image registration process is represented in the block diagram in Figure 5. First, the sensed color image is converted into gray-scale, and then it is transformed to the same scale of the reference image. Scaling is performed converting the sensed image to the resolution of the reference image. The scale factor s is calculated using (11)

$$\begin{pmatrix} s_x \\ s_y \end{pmatrix} = \begin{pmatrix} \frac{1}{f_x} \\ \frac{1}{f_y} \end{pmatrix} d \cdot I_{\text{res}}, \quad (11)$$

where d , as for the odometry, is the UAV ground altitude, and I_{res} is the resolution of the reference image. The alignment is performed using the UAV heading estimated by the KF.

After the alignment and scaling steps, the cross-correlation algorithm is applied. If S is the sensed image and I is the reference image, the expression for the two-dimensional normalized cross-correlation is

$$C(u, v) = \frac{\sum_{x,y} [S(x, y) - \mu_S][I(x - u, y - v) - \mu_I]}{\sqrt{\sum_{x,y} [S(x, y) - \mu_S]^2 \sum_{x,y} [I(x - u, y - v) - \mu_I]^2}}, \quad (12)$$

where μ_S and μ_I are the average intensity values of the sensed and the reference image, respectively. Figure 4 depicts a typical cross-correlation result between a sensed image taken from the Rmax helicopter and a restricted view of the reference image of the flight-test site.

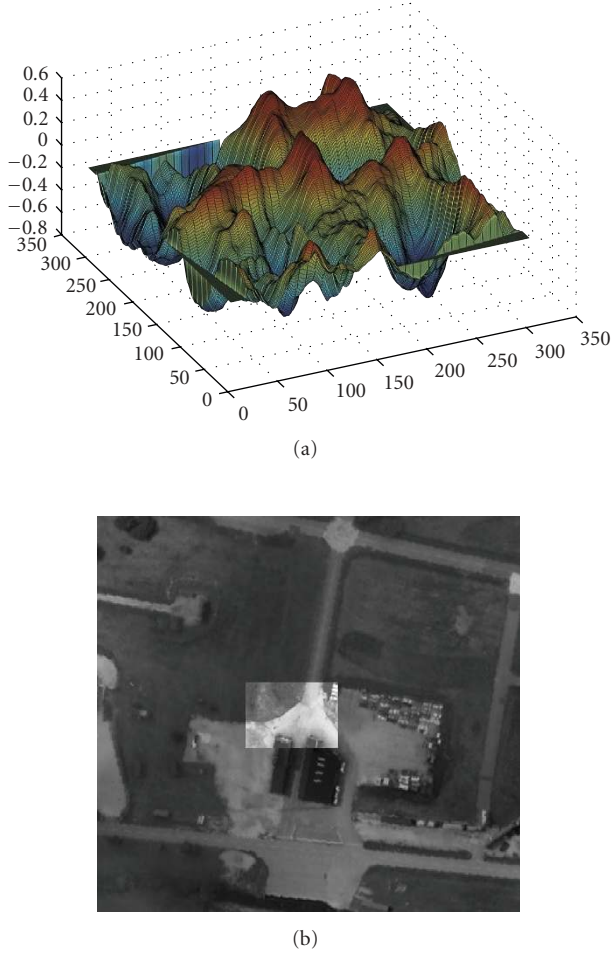


FIGURE 4: (a) depicts the normalized cross-correlation results, and (b) depicts the sensed image matched to the reference image.

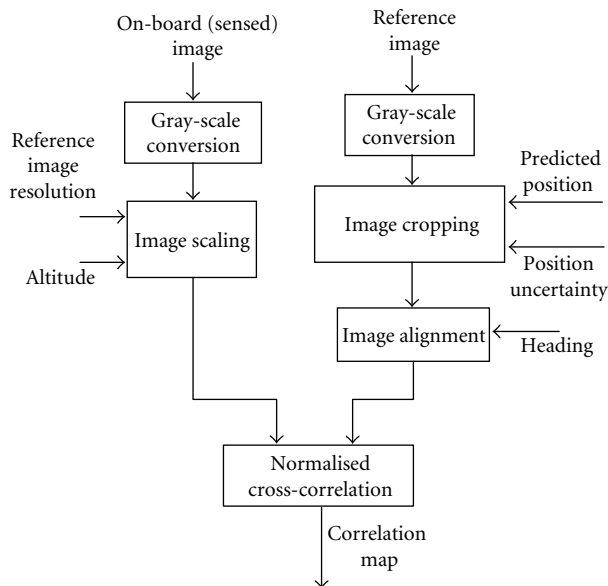


FIGURE 5: Image registration schematic.

Image registration is performed only on a restricted window of the reference image. The UAV position predicted by the filter is used as center of the restricted matching area. The purpose is to disregard low-probability areas to increase the computational efficiency of the registration process. The window size depends on the position uncertainty estimated by the filter.

5. Sensor Fusion Algorithms

In this work, the Bayesian estimation framework is used to fuse data from different sensor sources and estimate the UAV state. The state estimation problem has been split in two parts. The first part is represented by a standard Kalman filter (KF) which estimates the full UAV state (position, velocity, and attitude); the second part is represented by a point-mass filter (PMF) which estimates the absolute 2D UAV position. The two filters are interconnected as shown in Figure 2.

5.1. Bayesian Filtering Recursion. A nonlinear state-space model with additive process noise \vec{v} and measurement noise \vec{e} can be written in the following form:

$$\begin{aligned}\dot{\vec{x}}_t &= f(\vec{x}_{t-1}, \vec{u}_{t-1}) + \vec{v}, \\ \vec{y}_t &= h(\vec{x}_t, \vec{u}_t) + \vec{e}.\end{aligned}\quad (13)$$

If the state vector \vec{x} is n -dimensional, the recursive Bayesian filtering solution for such a model is

$$p(\vec{x}_t | \vec{y}_{1:t-1}) = \int_{\mathfrak{R}^n} p_v(\vec{x}_t - f(\vec{x}_{t-1}, \vec{u}_{t-1})) p(\vec{x}_{t-1} | \vec{y}_{1:t-1}) d\vec{x}_{t-1}, \quad (14)$$

$$\alpha_t = \int_{\mathfrak{R}^n} p_e(\vec{y}_t - h(\vec{x}_t, \vec{u}_t)) p(\vec{x}_t | \vec{y}_{1:t-1}) d\vec{x}_t, \quad (15)$$

$$p(\vec{x}_t | \vec{y}_{1:t}) = \alpha_t^{-1} p_e(\vec{y}_t - h(\vec{x}_t, \vec{u}_t)) p(\vec{x}_t | \vec{y}_{1:t-1}). \quad (16)$$

The aim of recursion (14)–(16) is to estimate the posterior density $p(\vec{x}_t | \vec{y}_{1:t})$ of the filter. Equation (14) represents the time update while (16) is the measurement update. p_v and p_e represent the PDFs of the process and measurement noise, respectively.

From the posterior $p(\vec{x}_t | \vec{y}_{1:t})$, the estimation of the minimum variance (MV) system state and its uncertainty are computed from (17) and (18), respectively, as follows:

$$\hat{\vec{x}}_t^{\text{MV}} = \int_{\mathfrak{R}^n} \vec{x}_t p(\vec{x}_t | \vec{y}_{1:t}) d\vec{x}_t, \quad (17)$$

$$P_t = \int_{\mathfrak{R}^n} (\vec{x}_t - \hat{\vec{x}}_t^{\text{MV}}) (\vec{x}_t - \hat{\vec{x}}_t^{\text{MV}})^T p(\vec{x}_t | \vec{y}_{1:t}) d\vec{x}_t. \quad (18)$$

In general, it is very hard to compute the analytical solution of the integrals in (14)–(18). If the state-space model (13) is linear, and the noises \vec{v} and \vec{e} are Gaussian and the prior $p(\vec{x}_0)$ normally distributed, the close form solution is represented by the popular Kalman filter.

The vision-based navigation problem addressed in this work is non-Gaussian, and therefore a solution based only on Kalman filter cannot be applied. The problem is non-Gaussian since the image registration likelihood does not have a Gaussian distribution (see Figure 4(a)).

The filtering solution explored in this work is represented by a standard Kalman filter which fuses inertial data with an absolute position measurement. The position measurement comes from a 2-state point-mass filter which fuses the visual odometry data with absolute position information coming from the image registration module (Figure 2). The approach has given excellent results during field trials, but it has some limitations as it will be shown later.

5.2. Kalman Filter. The KF implemented is the standard structure in airborne navigation systems and is used to estimate the error states from an integrated navigation system.

The linear state-space error dynamic model used in the KF is derived from a perturbation analysis of the motion equations [26] and is represented with the model (19)-(20) where system (19) represents the process model while system (20) represents the measurement model:

$$\begin{pmatrix} \delta \dot{r}^n \\ \delta \dot{v}^n \\ \dot{\epsilon}^n \end{pmatrix} = \begin{bmatrix} \mathbf{F}_{rr} & \mathbf{F}_{rv} & \mathbf{0} & & & \\ & & 0 & -a_d & a_e & \\ \mathbf{F}_{vr} & \mathbf{F}_{vv} & a_d & 0 & -a_n & \\ & & -a_e & a_n & 0 & \\ \mathbf{F}_{er} & \mathbf{F}_{ev} & & -(\vec{\omega}_{in}^n \times) & & \end{bmatrix} \begin{pmatrix} \delta r^{\vec{n}} \\ \delta v^{\vec{n}} \\ \epsilon^{\vec{n}} \end{pmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{R}_b^n & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}_b^n \end{bmatrix} \vec{u}, \quad (19)$$

$$\begin{pmatrix} \varphi_{ins} - \varphi_{vision} \\ \lambda_{ins} - \lambda_{vision} \\ h_{ins} - (\Delta h_{baro} + h_0) \end{pmatrix} = [\mathbf{I} \ \mathbf{0} \ \mathbf{0}] \begin{pmatrix} \delta r^{\vec{n}} \\ \delta v^{\vec{n}} \\ \epsilon^{\vec{n}} \end{pmatrix} + \vec{e} \quad (20)$$

where $\delta r^{\vec{n}} = (\delta\varphi \ \delta\lambda \ \delta h)^T$ are the latitude, longitude, and altitude errors; $\delta v^{\vec{n}} = (\delta v_n \ \delta v_e \ \delta v_d)^T$ are the north, east, and down velocity errors; $\epsilon^{\vec{n}} = (\epsilon_N \ \epsilon_E \ \epsilon_D)^T$ are the attitude errors in the navigation frame; $\vec{u} = (\delta \vec{f}_{acc}^T \ \delta \vec{\omega}_{gyro}^T)$ are the accelerometers and gyros noise; \vec{e} represents the measurement noise; $\vec{\omega}_{in}^n$ is the rotation rate of the navigation frame; a_n, a_e, a_d are the north, east and vertical acceleration components; \mathbf{F}_{xx} are the elements of the system's dynamics matrix.

The KF implemented uses 12 states including 3 accelerometer biases as mentioned before. However, the state-space model (19)-(20) uses only 9 states without accelerometer biases. The reason for using a reduced representation is to allow the reader to focus only on the relevant part of the model for this section. The accelerometer biases are modeled as first-order Markov processes. The gyros biases can be modeled in the same way, but they were not used in this implementation. The acceleration elements $a_n, a_e,$ and a_d are left in the explicit form in the system (19) as they will be needed for further discussions.

It can be observed how the measurements coming from the vision system in the form of latitude and longitude ($\varphi_{vision}, \lambda_{vision}$) are used to update the KF. The altitude measurement update is done using the barometric altitude information from an on-board pressure sensor. In order to compute the altitude in the WGS84 reference system required to update the vertical channel, an absolute reference altitude measurement h_0 needs to be known. For example if h_0 is taken at the take-off position, the barometric altitude variation Δh_{baro} relative to h_0 can be obtained from the pressure sensor, and the absolute altitude can finally be computed. This technique works if the environmental static pressure remains constant. The state-space system, as represented in (19), (20), is fully observable. The elements of the matrix $\mathbf{F}_{er}, \mathbf{F}_{ev}$ and $(\vec{\omega}_{in}^n \times)$ are quite small as they depend on the Earth rotation rate and the rotation rate of the navigation frame due to the Earth's curvature. These elements are influential in the case of a very sensitive IMU or high-speed flight conditions. For the flight conditions and typical IMU used in a small UAV helicopter, such elements are negligible, and therefore observability issues might arise for the attitude angles. For example, in case of hovering flight conditions or, more generally, in case of nonaccelerated horizontal flight conditions, the elements a_n and a_e are zero. It can also be observed that the angle error ϵ_D (third component of $\epsilon^{\vec{n}}$) is not observable. In case of small pitch and roll angles ϵ_D corresponds to the heading angle. Therefore, for helicopters that are supposed to maintain the hovering flight condition for an extended period of time, an external heading aiding (i.e., compass) is required. On the other hand, the pitch and roll angle are observable since the vertical acceleration a_d is usually different from zero (except in case of free fall but this is an extreme case).

It should be mentioned that the vision system uses the attitude angles estimated by the KF for the computation of the absolute position, as can be observed in Figure 2 (subsystem 2), where this can be interpreted as a linearization of the measurement update equation of the KF. This fact might have implications on the estimation of the attitude angles since an information loop is formed in the scheme. The issue will be discussed later in this paragraph.

5.3. Point-Mass Filter. The point-mass filter (PMF) is used to fuse measurements coming from the visual odometry system and the image matching system. The PMF computes the solution to the Bayesian filtering problem on a discretized grid [27]. In [2] such a technique was applied to a terrain navigation problem where a digital terrain elevation model was used instead of a digital 2D image as in the case presented here. The PMF is particularly suitable for this kind of problem since it handles general probability distributions and nonlinear models.

The problem can be represented with the following state-space model:

$$\vec{x}_t = \vec{x}_{t-1} + \vec{u}_{t-1}^{odom} + \vec{v}, \quad (21)$$

$$p(\vec{y}_t | \vec{x}_t, \vec{z}_t). \quad (22)$$

Equation (21) represents the process model where \vec{x} is the two-dimensional state (north-east), \vec{v} is the process noise, and \vec{u}^{odom} is the position displacement between time $t - 1$ and t computed from the visual odometry (\vec{u}^{odom} was indicated with \vec{t} in Section 3). For the process noise \vec{v} , a zero mean Gaussian distribution and a standard deviation value $\sigma = 2$ m are used. Such value has been calculated through a statistical analysis of the odometry through Monte Carlo simulations not reported in this paper.

The observation model is represented by the likelihood (22) and represents the probability of measuring \vec{y}_t given the state vectors \vec{x}_t and \vec{z}_t . The latter is a parameter vector given by $\vec{z}_t = (\hat{\phi} \ \hat{\theta} \ \hat{\psi} \ d)$. The first three components are the estimated attitude angles from the Kalman filter while d is the ground altitude measured from the barometric sensor. A certainty equivalence approximation has been used here since the components of vector \vec{z}_t are taken as deterministic values. The measurement likelihood (22) is computed from the cross-correlation (12), and its distribution is non-Gaussian. The distribution given by the cross-correlation (12) represents the correlation of the on-board camera view with the reference image. In general, there is an offset between the position matched on the reference image and the UAV position due to the camera view angle. The offset must be removed in order to use the cross-correlation as probability distribution for the UAV position. The attitude angles and ground altitude are used for this purpose.

As discussed earlier, in the PMF approximation the continuous state-space is discretized over a two-dimensional limited size grid, and so the integrals are replaced with finite sums over the grid points. The grid used in this application is uniform with N number of points and resolution δ . The Bayesian filtering recursion (14)–(18) can be approximated as follows:

$$p(\vec{x}_t(k) | \vec{y}_{1:t-1}) = \sum_{n=1}^N p_v(\vec{x}_t(k) - \vec{u}_{t-1}^{\text{odom}} - \vec{x}_{t-1}(n)) \quad (23)$$

$$\times p(\vec{x}_{t-1}(n) | \vec{y}_{1:t-1}) \delta^2,$$

$$\alpha_t = \sum_{n=1}^N p(\vec{y}_t(n) | \vec{x}_t(n), \vec{z}_t) p(\vec{x}_t(n) | \vec{y}_{1:t-1}) \delta^2, \quad (24)$$

$$p(\vec{x}_t(k) | \vec{y}_{1:t}) = \alpha_t^{-1} p(\vec{y}_t(n) | \vec{x}_t(n), \vec{z}_t) p(\vec{x}_t(k) | \vec{y}_{1:t-1}), \quad (25)$$

$$\hat{\vec{x}}_t^{\text{MV}} = \sum_{n=1}^N \vec{x}_t(n) p(\vec{x}_t(n) | \vec{y}_{1:t}) \delta^2, \quad (26)$$

$$P_t = \sum_{n=1}^N \left(\vec{x}_t(n) - \hat{\vec{x}}_t^{\text{MV}} \right) \left(\vec{x}_t(n) - \hat{\vec{x}}_t^{\text{MV}} \right)^T p(\vec{x}_t(n) | \vec{y}_{1:t}) \delta^2. \quad (27)$$

Before computing the time update (23), the grid points are translated according to the displacement calculated from

the odometry. The convolution (23) requires N^2 operations and computationally is the most expensive operation of the recursion. In any case, due to the problem structure, the convolution has separable kernels. For this class of two-dimensional convolutions there exist efficient algorithms which reduce the computational load. More details on this problem can be found in [2].

Figure 6 depicts the evolution of the filter probability density function (PDF). From the prior (t_0), the PDF evolves developing several peaks. The images and flight data used were collected during a flight-test campaign, and for the calculation a 200×200 grid of 1 meter resolution was used.

The interface between the KF and the PMF is realized by passing the latitude and longitude values computed from the PMF to the KF measurement equation (20). The PMF position covariance is computed with (27), and it could be used in the KF for the measurement update. However, the choice of an empirically determined covariance for the KF measurement update has been preferred. The constant values of 2 meters for the horizontal position uncertainty and 4 meters for the vertical position uncertainty (derived from the barometric altitude sensor) have been used.

5.4. Stability Analysis from Monte Carlo Simulations.

In this section the implications of using the attitude angles estimated by the KF to compute the vision measurements will be analyzed. As mentioned before, since the vision measurements are used to update the KF, an information loop is created which could limit the performance and stability of the filtering scheme. First, the issue will be analyzed using a simplified dynamic model. Then, the complete KF in close loop with the odometry will be tested using Monte Carlo simulations. The velocity error dynamic model implemented in the KF is derived from a perturbation of the following velocity dynamic model [28]:

$$\dot{\vec{v}}^n = R_b^n \vec{a}^b - (2\vec{\omega}_{ie}^n + \vec{\omega}_{en}^n) \times \vec{v}^n + \vec{g}^n \quad (28)$$

with \vec{a}^b representing the accelerometers output, $\vec{\omega}_{ie}^n$ and $\vec{\omega}_{en}^n$ the Earth rotation rate and the navigation rotation rates which are negligible, as discussed in Section 2, and \vec{g}^n the gravity vector. Let us analyze the problem for the 1D case. The simplified velocity dynamics for the x^n axis is

$$\dot{v}_x^n = a_x^b \cos \theta + a_z^b \sin \theta, \quad (29)$$

where θ represents the pitch angle. Suppose that the helicopter is in hovering but the estimated pitch angle begins to drift due to the gyro error. An apparent velocity will be observed due to the use of the attitude angles in the measurement equation from the vision system. Linearizing the model around the hovering condition gives for the velocity dynamics and observation equations:

$$\dot{v}_x^n = a_x^b + a_z^b \theta, \quad (30)$$

$$h \dot{\theta} = v_x^n, \quad (31)$$

where h is the ground altitude measured by the barometric sensor. The coupling between the measurement equation

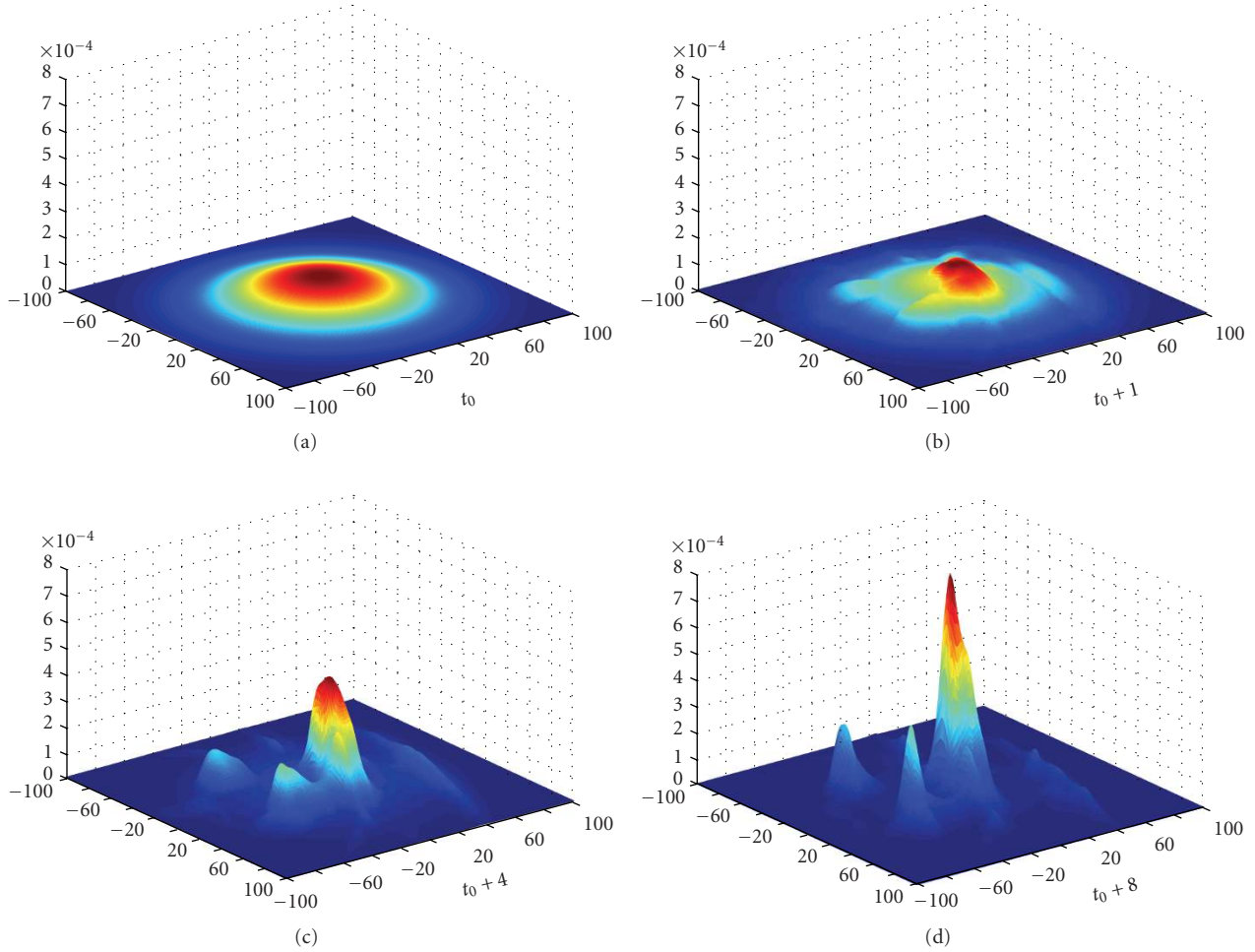


FIGURE 6: Evolution of the filter's probability density function (PDF). The capability of the filter to maintain the full probability distribution over the grid space can be observed.

(31) with the dynamic equation (30) can be observed (θ is in fact used to disambiguate the translation from the rotation in the measurement equation). Substituting (31) in (30) gives

$$h\ddot{\theta} = a_x^b + a_z^b\theta. \quad (32)$$

Equation (32) is a second-order differential equation similar to the equation for a spring-mass dynamic system (a_z^b has a negative value) where the altitude h plays the role of the mass. One should expect that the error introduced by the information loop has an oscillatory behavior which changes with the change in altitude. Since this is an extremely simplified representation of the problem, the behavior of the Kalman filter in close loop with the odometry was tested through Monte Carlo simulations.

Gaussian-distributed accelerometer and gyro data were generated together with randomly distributed features in the images. Since the analysis has the purpose of finding the effects of the attitude coupling problem, the following procedure was applied. First, Kalman filter results were obtained using the noisy inertial data and constant position measurement update. Then, the KF was fed with the same inertial data as the previous case but the measurement

update came from the odometry instead (the features in the image were not corrupted by noise since the purpose was to isolate the coupling effect).

The KF results for the first case (constant update) were considered as the reference, and then the results for the second case were compared to the reference one. What is shown in the plots is the difference of the estimated pitch angle (Figure 7) and roll angle (Figure 8) between the two cases at different altitudes (obviously the altitude influences only the results of the closed loop case).

For altitudes up to 100 meters the difference between the two filter configurations is less than 0.5 degree. However, the increasing of the flight altitude makes the oscillatory behavior of the system more evident with an increasing in amplitude (as expected from the previous simplified analysis). A divergent oscillatory behavior was observed from an altitude of about 700 meters.

This analysis shows that the updating method used in this work introduces an oscillatory dynamics in the filter state estimation. The effect of such dynamics has a low impact for altitudes below 100 meters while becomes more severe for larger altitudes.

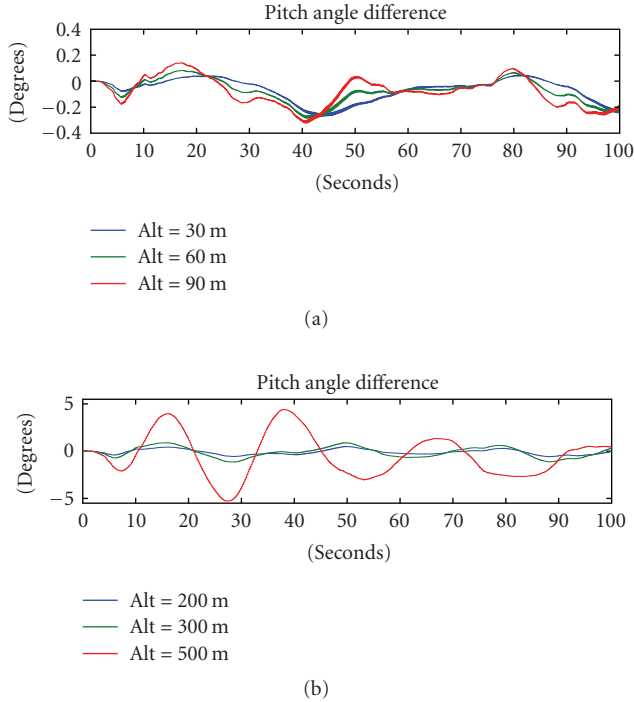


FIGURE 7: Difference between the pitch angle estimated by the KF with a constant position update and the KF updated using odometry position. It can be noticed how the increasing of the flight altitude makes the oscillatory behavior of the system more evident with an increasing in amplitude.

6. UAV Platform

The proposed filter architecture has been tested on real flight-test data and on-board autonomous UAV helicopter. The helicopter is based on a commercial Yamaha Rmax UAV helicopter (Figure 1). The total helicopter length is about 3.6 m. It is powered by a 21 hp two-stroke engine, and it has a maximum take-off weight of 95 kg. The avionic system was developed at the Department of Computer and Information Science at Linköping University and has been integrated in the Rmax helicopter. The platform developed is capable of fully autonomous flight from take-off to landing. The sensors used in this work consist of an inertial measurement unit (three accelerometers and three gyros) which provides the helicopter's acceleration and angular rate along the three body axes, a barometric altitude sensor, and a monocular CCD video camera mounted on a pan/tilt unit. The avionic system is based on 3 embedded computers. The primary flight computer is a PC104 PentiumIII 700 MHz. It implements the low-level control system which includes the control modes (take-off, hovering, path following, landing, etc.), sensor data acquisition, and the communication with the helicopter platform. The second computer, also a PC104 PentiumIII 700 MHz, implements the image processing functionalities and controls the camera pan-tilt unit. The third computer, a PC104 Pentium-M 1.4 GHz, implements high-level functionalities such as path-planning and task-planning. Network communication between computers is

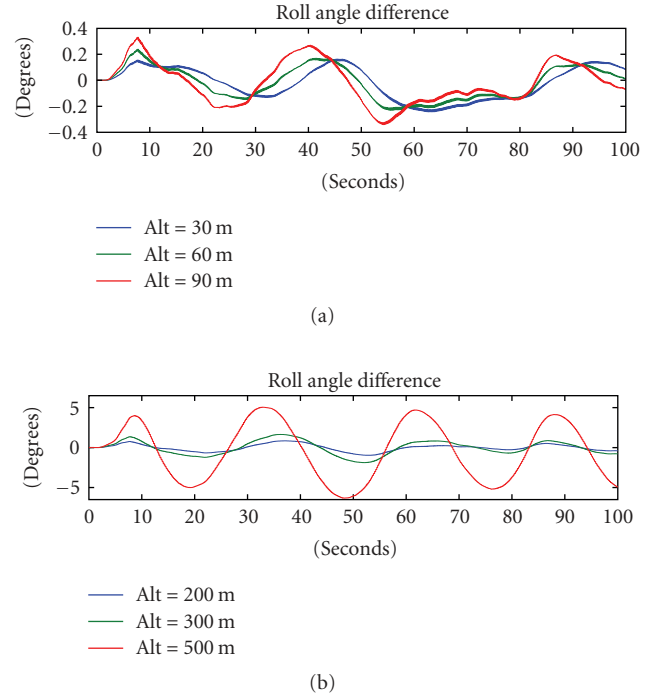


FIGURE 8: Difference between the roll angle estimated by the KF with a constant position update and the KF updated using odometry position. It can be noticed how the increasing of the flight altitude makes the oscillatory behavior of the system more evident with an increasing in amplitude.

physically realized with serial line RS232C and Ethernet. Ethernet is mainly used for remote login and file transfer, while serial lines are used for hard real-time networking.

7. Experimental Results

In this section the performance of the vision-based state estimation approach described will be analyzed. Experimental evaluations based on offline real flight data as well as online on-board test results are presented. The flight-tests were performed in an emergency services training area in the south of Sweden.

The reference image of the area used for this experiment is an orthorectified aerial image of 1 meter/pixel resolution with a submeter position accuracy. The video camera sensor is a standard CCD analog camera with approximately 45 degrees horizontal angle of view. The camera frame rate is 25 Hz, and the images are reduced to half resolution (384×288 pixels) at the beginning of the image processing pipeline to reduce the computational burden. During the experiments, the video camera was looking downward and fixed with the helicopter body. The PMF recursion was computed in all experiments on a 80×80 meters grid of one meter resolution. The IMU used is provided by the Yamaha Motor Company and integrated in the Rmax platform. Table 1 provides available specification of the sensors used in the experiment.

TABLE 1: Available characteristics of the sensor used in the navigation algorithm.

Sensor	Output rate	Resolution	Bias
Accelerometers	66 Hz	1 mG	13 mg
Gyros	200 Hz	0.1 deg/s	<0.1 deg/s
Barometer	40 Hz	0.1 m	—
Vision	25 Hz	384 × 288 pixels	—

The results of the vision-based navigation algorithm are compared to the navigation solution given by an on-board INS/GPS KF running on-line. The KF fuses the inertial sensors with GPS position data and provides the full helicopter state estimate. The position data used as reference are provided by a real-time kinematic GPS receiver with a submeter position accuracy.

7.1. Performance Evaluation Using Offline Flight Data. Sensor data and on-board video were recorded during an autonomous flight. The vision-based filter is initialized with the last valid state from the INS/GPS filter, and after the initialization the GPS is not used anymore. The position update to the KF is then provided by the vision system. The inertial data are sampled and integrated at a 50 Hz rate, while the vision system provides position update at 4 Hz rate.

Figure 9 shows the UAV flight path reconstructed using the navigation approach described in this work without using the GPS. The helicopter flew a closed loop path of about 1 kilometer length at 60 meters constant altitude above the ground. Figure 9(a) presents a comparison between the helicopter path computed by the vision-based system (PMF output) with the GPS reference. The vision-based position is always close to the GPS position indicating a satisfactory localization performance. The odometry position results are also displayed in Figure 9(a). Figure 9(b) shows the position uncertainty squared along the path estimated from the PMF (27) and graphically represented with ellipses. The ellipse axes are oriented along the north-east directions (the uncertainty is in fact estimated along such axes by the PMF). The uncertainty would be better represented with ellipses oriented along the direction of maximum and minimum uncertainty. In any case it is interesting to notice that above a road segment the uncertainty increases along the road and decreases along the direction perpendicular to the road. This can be noticed along the first path's leg (between points 1 and 2) and at point 4 of the path. This fact is a consequence of the impossibility for the image registration module to find the proper matching location along the road direction. It can also be noticed how the uncertainty increases when the helicopter passes above the pond (leg 3-4). This is a sign that this area does not have good properties for image registration. The position uncertainty was initialized to 25 meters at the beginning of the path (point 1).

Figures 10(a) and 10(b) show the north and east vision-based KF velocity estimation while the Figure 10(d) shows the horizontal velocity error magnitude. Figure 10(c) shows

the horizontal position error magnitude of the PMF, visual odometry and stand alone inertial solution. It can be observed how the PMF position error is always below 8 meters during the flight while the odometry accumulates 25 meters error in about 1 km of flight path length. The inertial solution has a pronounced drift showing the low-performance characteristics of the IMU used and giving an idea of the improvement introduced just by the odometry alone.

Figure 11 shows the attitude angles estimated by the KF (left column) and the attitude angle errors (right column). From the pitch and roll error plots there, does not appear to be any tendency to drift during the flight. The heading error plot shows instead a tendency to drift away. The low acceleration of the helicopter during the flight experiment makes the heading angle weakly observable (Section 2), therefore the use of an external heading information (i.e., compass) is required in the KF in order to allow for a robust heading estimate.

In Figure 12 the influence of the number of features tracked and the effects of the RANSAC algorithm are analyzed. The plots in the left column are relative to the case of the KLT tracking a maximum number of 50 features per frame while the right column shows the case of KLT tracking a maximum number of 150 features. When the maximum number of features is set, the tracking algorithm tries to track them, but usually a number of features will be lost for each frame so the number of features tracked are less than the maximum number. In addition, when the RANSAC algorithm is running, an additional number of features are discarded because they are considered outliers. In Figure 12(a), the odometry error is shown in the cases with and without the use of RANSAC and with a maximum of 50 features tracked. It can be observed how the use of RANSAC does not improve the odometry drift rate, but it helps to filter out the position jumps. It has been observed that the jumps occur during sudden helicopter attitude variation resulting in a large feature displacement in the image. Such a correlation can be observed comparing the pitch plot with the odometry error plot. As the images were sampled at a 4Hz rate, the problem could be mitigated by increasing the feature tracking frame rate. The figure also shows the number of features tracked and discarded from the RANSAC algorithm. In Figure 12(b) the same test is done with a maximum of 150 features tracked in each frame. If Figures 12(a) and 12(b) are compared, it can be observed that increasing the number of features tracked produces a worse result (comparing the blue plots without RANSAC). This effect is somehow predictable as the KLT selects the feature quality in a decreasing order. Comparing the two cases with RANSAC (red plots) there are basically no relevant differences meaning that the RANSAC detects the great part of outliers introduced by tracking a larger number of features (compare the two plots at the bottom of Figure 12). The fact that the cases without RANSAC finish with a lower position error does not have any relevance as it is mostly a random effect. From these considerations the solution with 50 features and RANSAC is preferable.

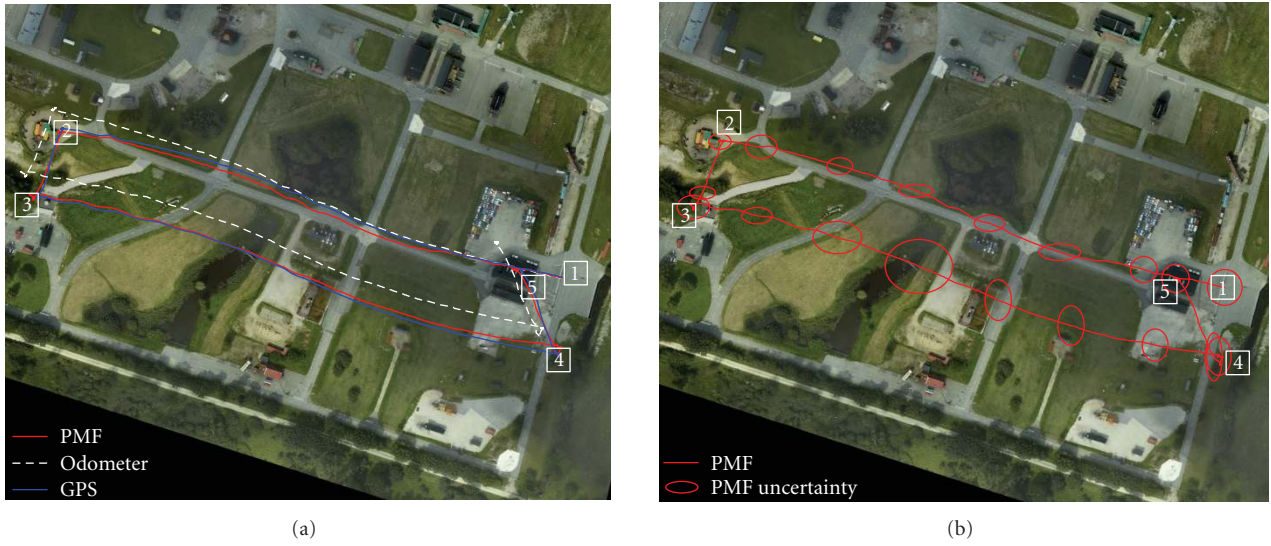


FIGURE 9: (a) displays the comparison between the flight path computed with the point-mass filter (red), the odometry (dashed white), and the GPS reference (blue). (b) shows the PMF result with the position uncertainty squared represented with ellipses along the path.

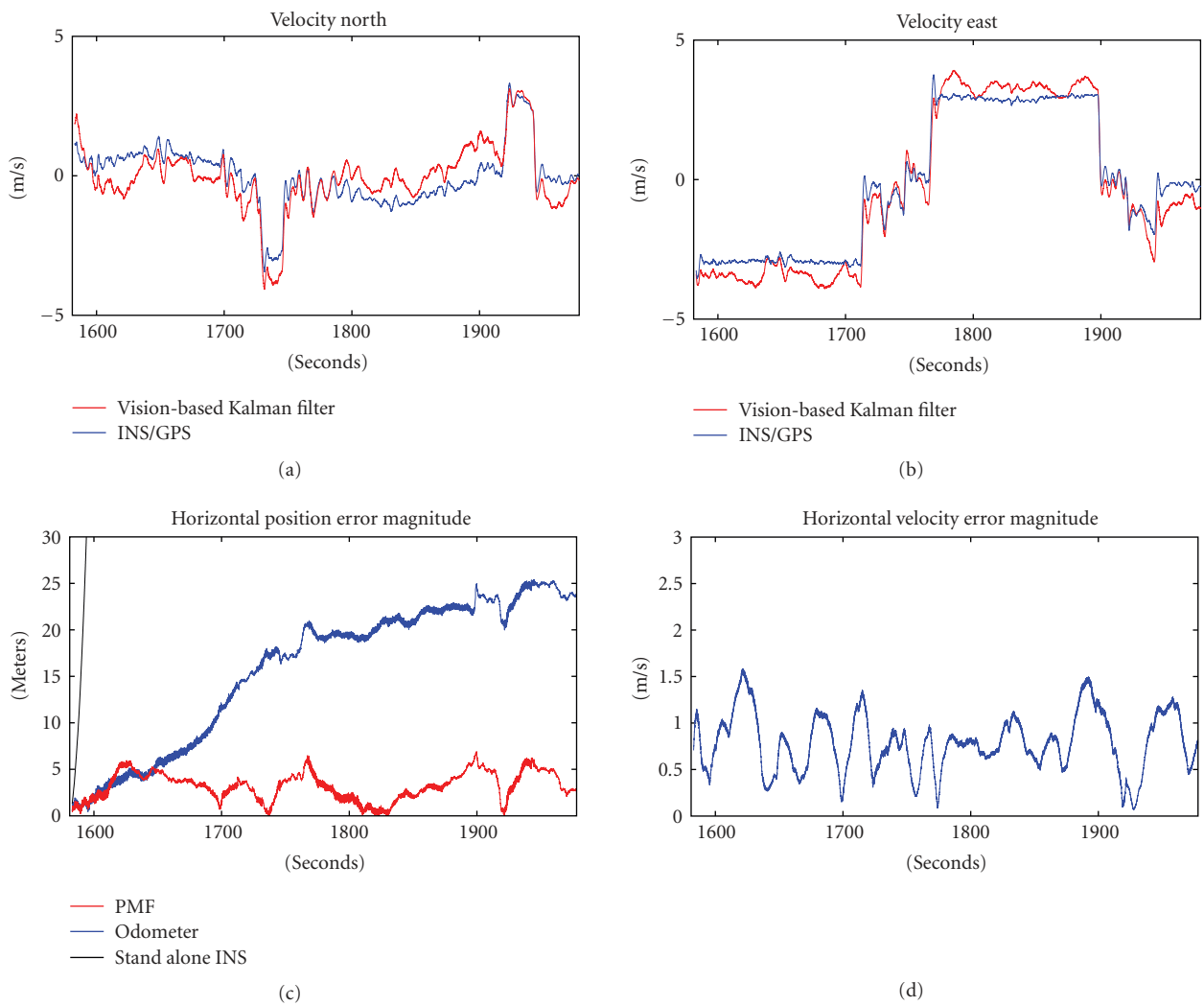


FIGURE 10: (a) and (b) depict the north and east velocity components estimated by the Kalman filter while (d) depicts the horizontal velocity error magnitude. In (c) the position error comparison between PMF, odometry, and stand alone INS is given.

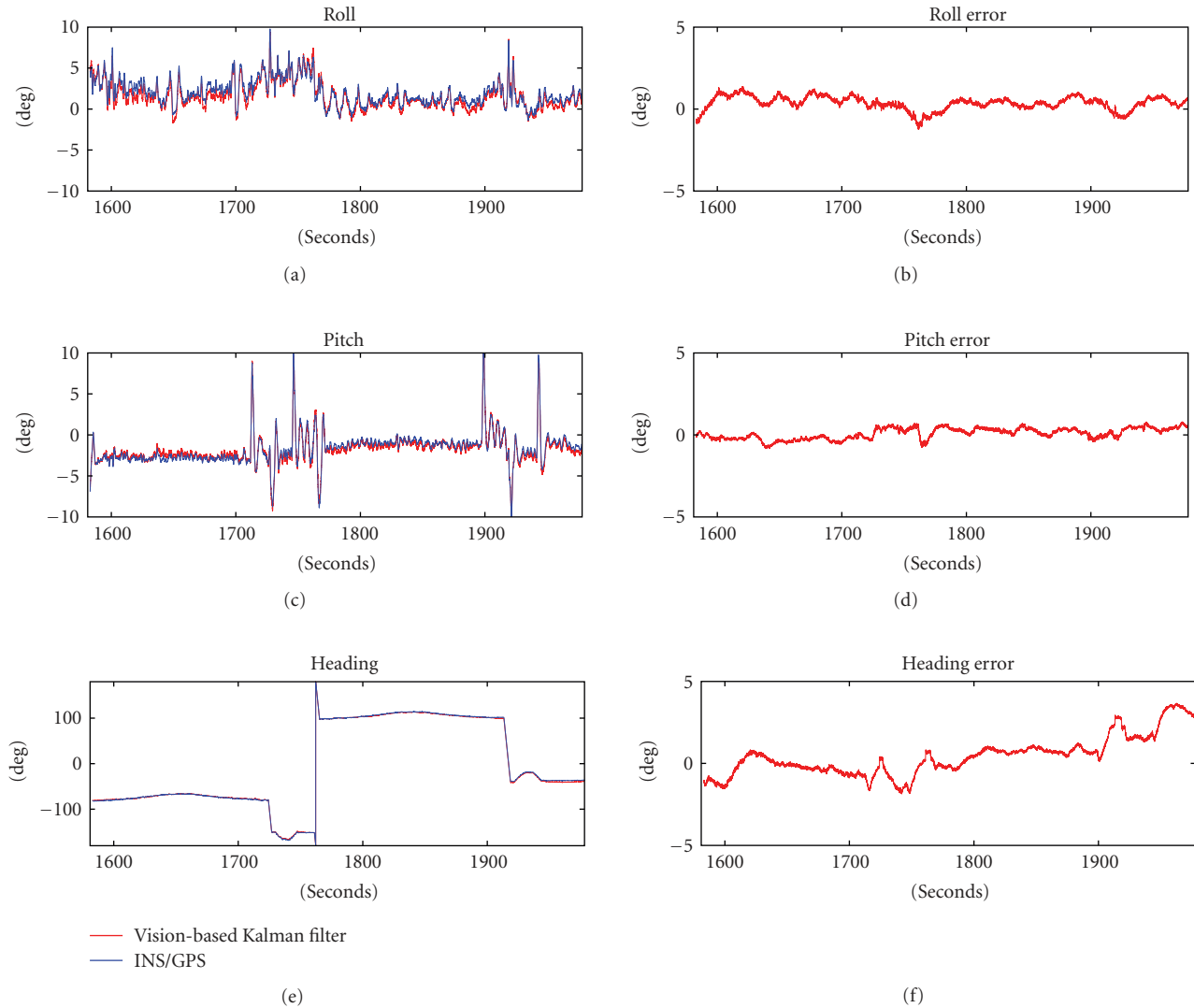


FIGURE 11: Estimated Kalman filter attitude angles with error plots.

Figures 13 and 14 show the results based on the same flight data set, but the vision-based navigation system has been initialized with 5 degrees error in pitch, roll, and heading angles. It can be observed in Figure 14 how the pitch and roll converge rapidly to the right estimate (confirming the assumptions and results demonstrated in Section 5) while the heading shows a tendency to converge in the long term, but practically the navigation algorithm was between 5 and 10 degrees off-heading during the whole test. It is interesting to notice from Figure 13 the effects of the off-heading condition. The odometry is rotated by approximately the heading angle error. The PFM solution is degraded due to the increased odometry error and due to the fact that there is an image misalignment error between the sensed and reference images. Despite the image misalignment, the PMF position estimate closes the loop at the right location (point 5) thanks to the robustness of the image cross-correlation with respect to small image misalignment. It appears that the

image correlation was especially robust around corner 4 of the path. This fact is verified by the sudden decrease of the position uncertainty which can be observed in Figure 13.

7.2. Real-Time on-Board Flight-Test Results. Flight-test results of the vision-based state estimation algorithm implemented on-board the Rmax helicopter platform will be presented here. The complete navigation architecture is implemented on two on-board computers in the C language. The Subsystem 1 (Figure 2) is implemented on the primary flight computer PFC (PC104 PentiumIII 700 MHz) and runs at a 50 Hz rate. The Subsystem 2 is implemented on the image processing computer IPC (also a PC104 PentiumIII 700 MHz) and runs at about 7 Hz rate. The image processing is implemented using the Intel OpenCV library. The real-time data communication between the two computers is realized using a serial line RS232C. The data sent from

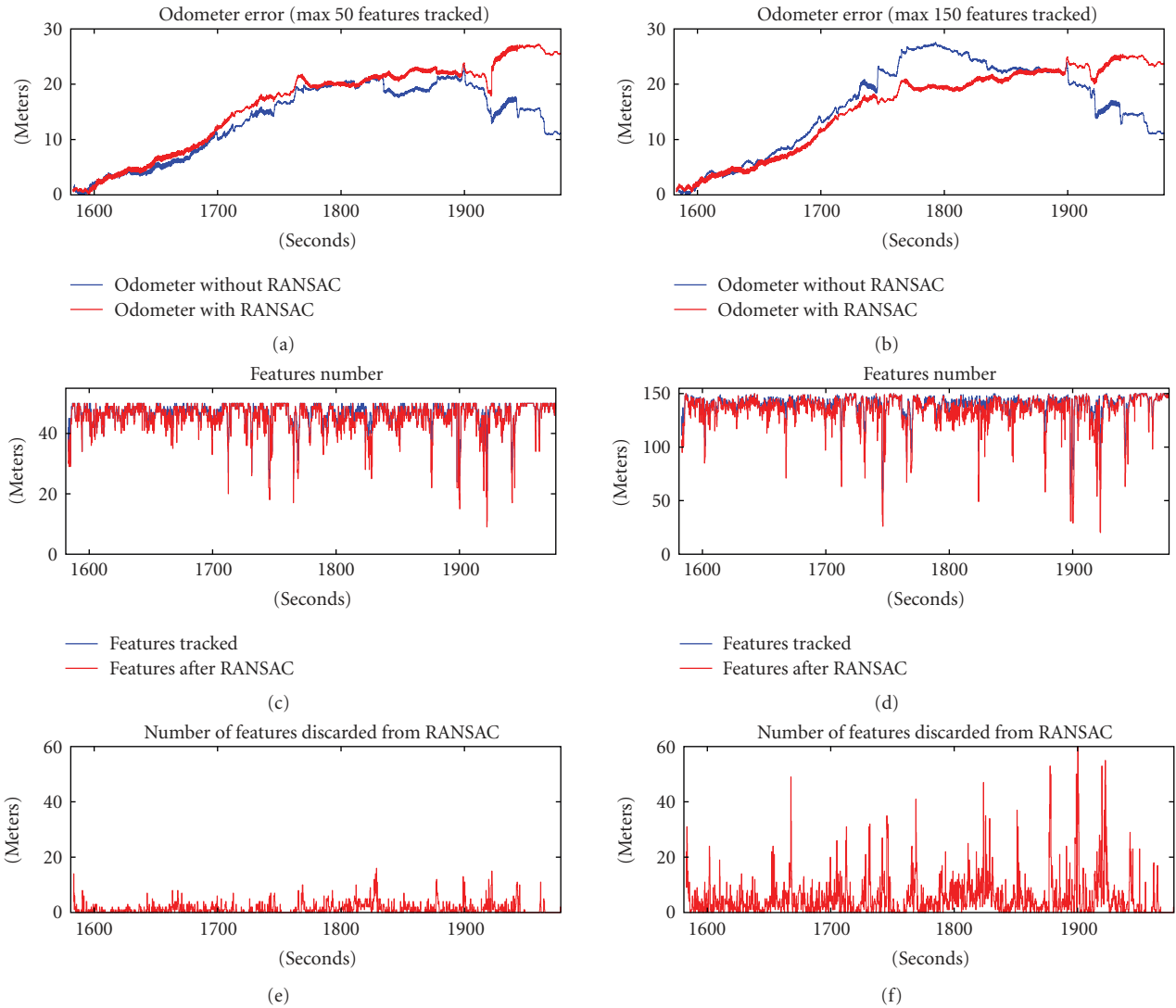


FIGURE 12: Comparison between odometry calculation using maximum 50 features (left column) and maximum 150 features (right column). The comparison also shows the effects of the RANSAC algorithm on the odometry error.

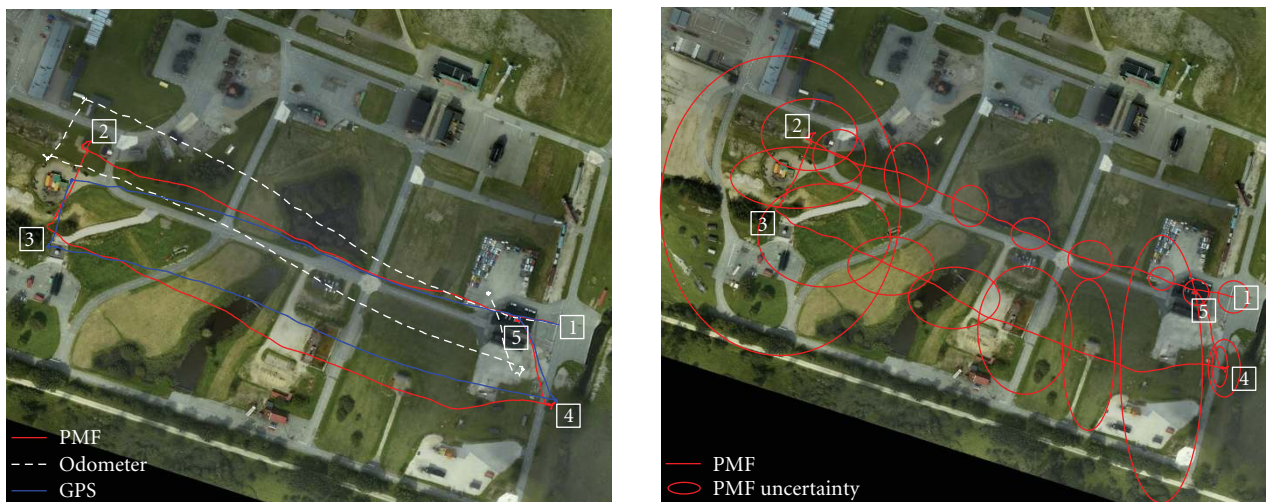


FIGURE 13: Vision-based navigation algorithm results in off-heading error conditions. The navigation filter is initialized with a 5-degree error in pitch, roll, and heading. While the pitch and roll angle converge to the right estimate rapidly, the heading does not converge to the right estimate affecting the odometry and the image cross-correlation performed in off-heading conditions.

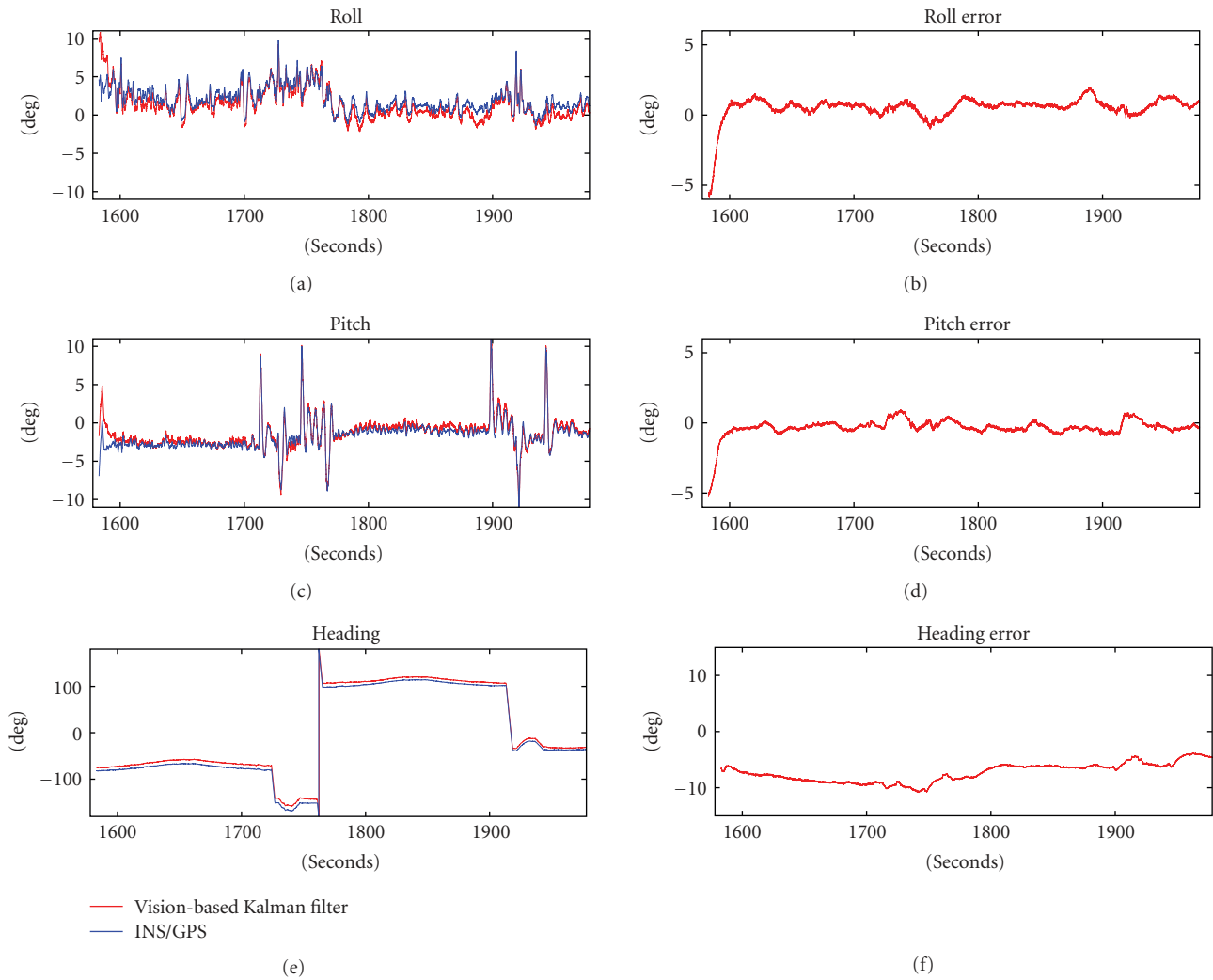


FIGURE 14: Estimated Kalman filter attitude angles with an initial 5-degree error added for roll, pitch, and heading angles. It can be observed how the pitch and roll angles converge quite rapidly, while the heading shows a weak converge tendency.

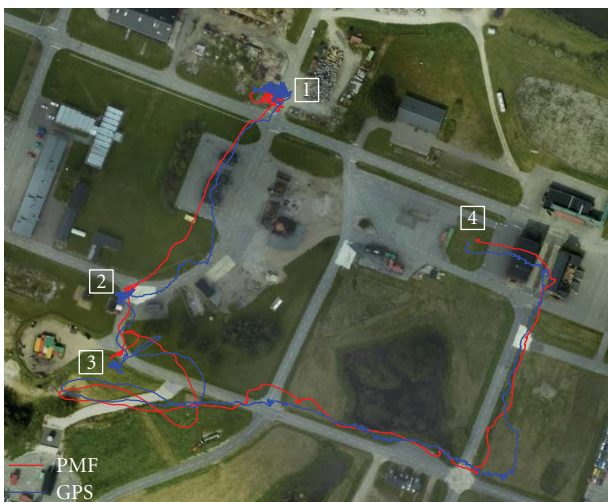


FIGURE 15: Real-time on-board position estimation results. Comparison between the flight path computed with the PMF (red) and the GPS reference (blue).

the PFC to the IPC is the full vision-based KF state while the data sent from the IPC to the PFC is the latitude and longitude position estimated by the PMF and used to update the KF.

The on-board flight-test results are displayed in Figures 15, 16, and 17. The odometry ran with a maximum number of 50 features tracked in the image without RANSAC as it was not implemented on-board at the time of the experiment. The helicopter was manually put in hovering mode at an altitude of about 55 meters above the ground (position 1 of Figure 15). Then the vision-based navigation algorithm was initialized from the ground station. Subsequently the helicopter from manual was switched into autonomous flight with the control system taking the helicopter state from the vision-based KF. The helicopter was commanded to flight from position 1 to position 2 along a straight line path. Observe that during the path segment 1-2, the vision-based solution (red line) resembles a straight line as the control system was controlling using the vision-based data. The real path taken by the helicopter is instead

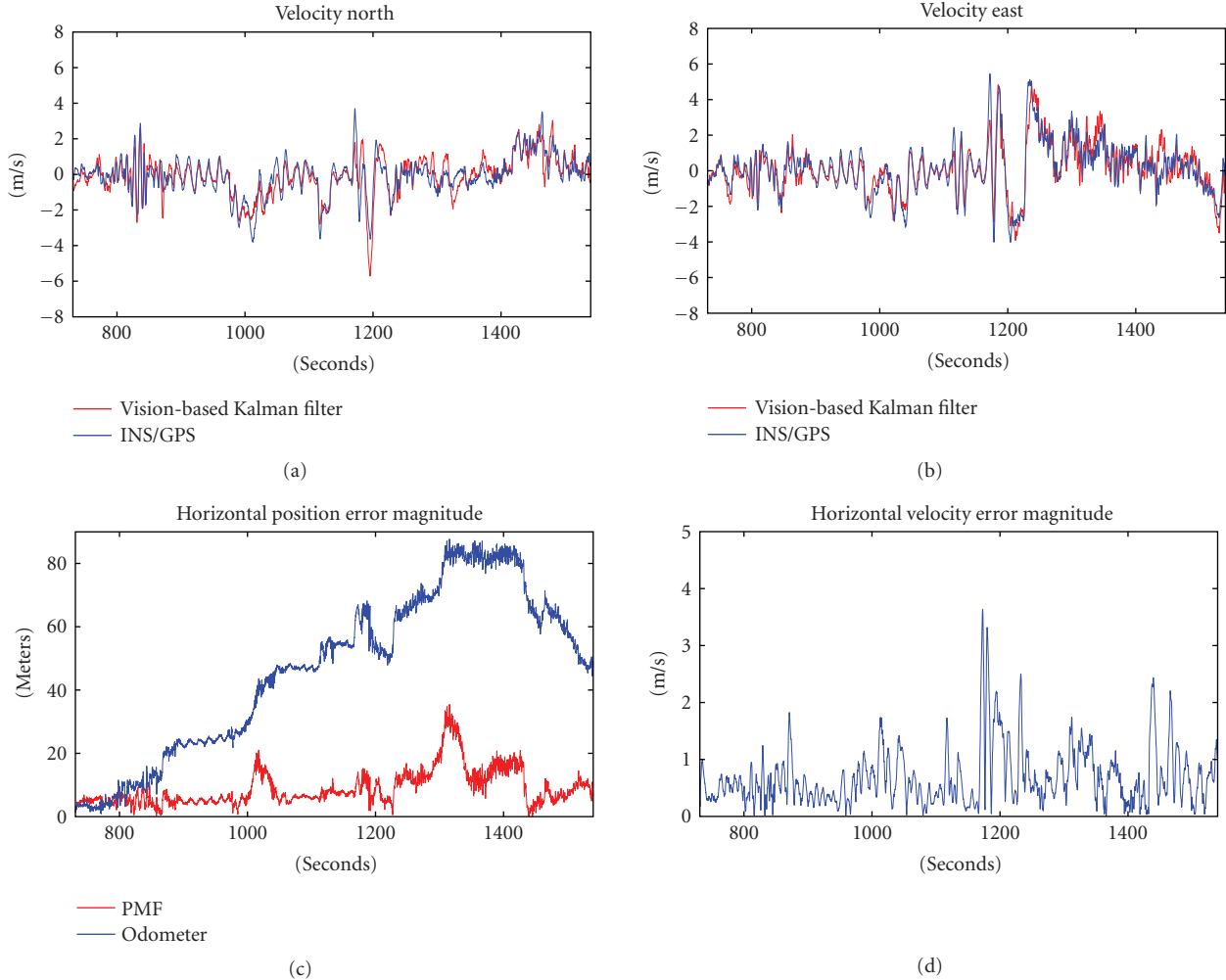


FIGURE 16: Real-time on-board results. (a) and (b) show the north and east velocity components estimated by the Kalman filter while (d) displays the horizontal velocity error magnitude. In (c) the position error comparison between PMF and odometry is given.

the blue one (GPS reference). The helicopter was then commanded to fly a spline path (segment 2-3). With the helicopter at the hovering point 3 the autonomous flight was aborted, and the helicopter was taken into manual flight. The reason the autonomous flight was aborted was that in order for the helicopter to exit the hovering condition and enter a new path segment the hovering stable condition must be reached. This is a safety check programmed in a low-level state machine which coordinates the flight mode switching. The hovering stable condition is fulfilled when the helicopter speed is less than 1 m/s. As can be seen from the velocity error plot in Figure 16 this is a rather strict requirement which is at the border of the vision-based velocity estimation accuracy. In any case, the vision-based algorithm was left running on-board while the helicopter was flown manually until position 4. The vision-based solution was always rather close to the GPS position during the whole path. Even the on-board solution confirms what was observed during the offline tests for the attitude angles (Figure 17) with a stable pitch and roll estimate and a nonstable heading estimate. Considering

the error in the heading estimate the PMF position results (Figure 15) confirm a certain degree of robustness of the image matching algorithm with respect to the image misalignment error.

8. Conclusions and Future Work

The experimental results of this work confirm the validity of the approach proposed. A vision-based sensor fusion architecture which can cope with short and long term GPS outages has been proposed and tested on-board unmanned helicopter. Since the performance of the terrain matching algorithms depends on the structural properties of the terrain, it could be wise to classify beforehand the reference images according to certain navigability criteria. In [29–31], methods for selecting scene matching areas suitable for terrain navigation are proposed.

In the future, a compass will be used as an external heading aid to the KF which should solve the heading estimation problem encountered here.

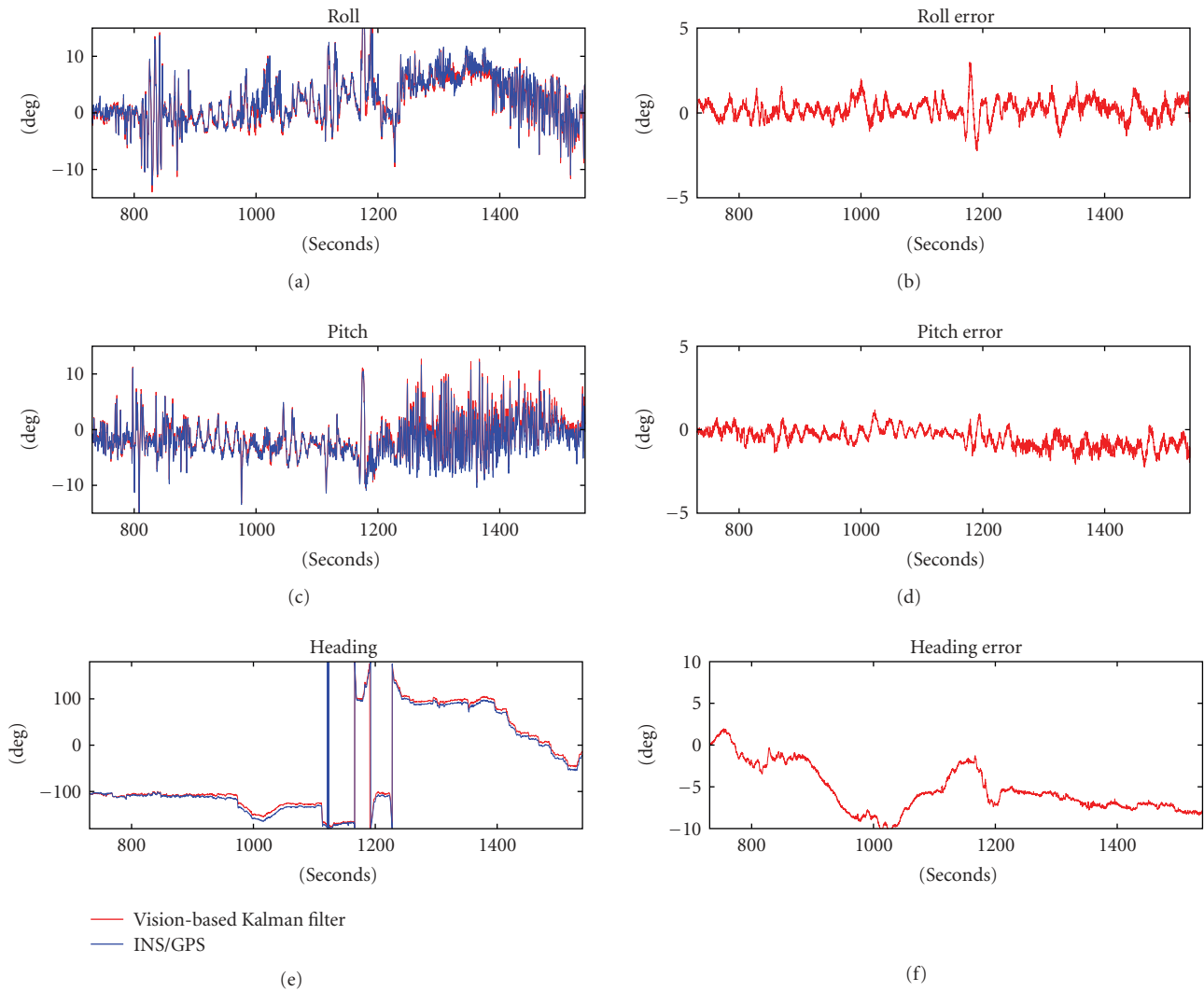


FIGURE 17: Real-time on-board results. Estimated Kalman filter attitude angles with error plots.

Acknowledgment

The authors would like to acknowledge their colleagues Piero Petitti, Mariusz Wzorek, and Piotr Rudol for the support given to this work. This work is supported in part by the National Aeronautics Research Program NFFP04-S4202, the Swedish Foundation for Strategic Research (SSF) Strategic Research Center MOVIII, the Swedish Research Council Linnaeus Center CADICS and LinkLab - Center for Future Aviation Systems.

References

- [1] C. James, et al., "Vulnerability assessment of the transportation infrastructure relying on the global positioning system," Tech. Rep., Volpe National Transportation Systems Center, US Department of Transportation, August 2001.
- [2] N. Bergman, *Recursive Bayesian estimation, navigation and tracking applications*, Ph.D. dissertation, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1999.
- [3] C. Tomasi and T. Kanade, "Detection and tracking of point features," Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, Pittsburgh, Pa, USA, April 1991.
- [4] D.-G. Sim, R.-H. Park, R.-C. Kim, S. U. Lee, and I.-C. Kim, "Integrated position estimation using aerial image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 1–18, 2002.
- [5] R. Karlsson, T. B. Schön, D. Törnqvist, G. Conte, and F. Gustafsson, "Utilizing model structure for efficient simultaneous localization and mapping for a UAV application," in *Proceedings of the IEEE Aerospace Conference*, Big Sky, Mont, USA, March 2008.
- [6] J. Kim and S. Sukkarieh, "Real-time implementation of airborne inertial-SLAM," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 62–71, 2007.
- [7] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, "Vision-based SLAM: stereo and monocular approaches," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343–364, 2007.
- [8] A. E. Johnson and J. F. Montgomery, "Overview of terrain relative navigation approaches for precise lunar landing," in *Proceedings of the IEEE Aerospace Conference*, Big Sky, Mont, USA, March 2008.

- [9] P. Maybeck, *Stochastic Models, Estimation and Control: Volume 1*, Academic Press, New York, NY, USA, 1979.
- [10] M. Svensson, *Aircraft trajectory restoration by integration of inertial measurements and GPS*, M.S. thesis, Linköping University, Linköping, Sweden, 1999.
- [11] E.-H. Shin, *Accuracy improvement of low cost INS/GPS for land applications*, M.S. thesis, University of Calgary, Calgary, Canada, 2001.
- [12] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1991.
- [13] O. Amidi, *An autonomous vision-guided helicopter*, Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, Pa, USA, 1996.
- [14] N. Frietsch, O. Meister, C. Schlaile, J. Seibold, and G. Trommer, "Vision based hovering and landing system for a vtol-mav with geo-localization capabilities," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, USA, August 2008.
- [15] S. S. Mehta, W. E. Dixon, D. MacArthur, and C. D. Crane, "Visual servo control of an unmanned ground vehicle via a moving airborne monocular camera," in *Proceedings of the American Control Conference*, pp. 5276–5281, Minneapolis, Minn, USA, June 2006.
- [16] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "A visual odometer without 3D reconstruction for aerial vehicles. Applications to building inspection," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4673–4678, Barcelona, Spain, 2005.
- [17] R. Hartley and A. Zisserman, *Multiple View Geometry*, Cambridge University Press, Cambridge, UK, 2nd edition, 2003.
- [18] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [19] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94)*, pp. 593–600, Seattle, Wash, USA, June 1994.
- [20] E. Michaelsen, M. Kirchhoff, and U. Stilla, "Sensor pose inference from airborne videos by decomposing homography estimates," in *International Archives of Photogrammetry and Remote Sensing*, M. O. Altan, Ed., vol. 35, pp. 1–6, 2004.
- [21] D. B. Barber, J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Vision-based target geo-location using a fixed-wing miniature air vehicle," *Journal of Intelligent and Robotic Systems*, vol. 47, no. 4, pp. 361–382, 2006.
- [22] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, no. 24, pp. 326–376, 1992.
- [23] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.
- [24] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [25] N. Trawny, A. I. Mourikis, S. I. Roumeliotis, et al., "Coupled vision and inertial navigation for pin-point landing," in *NASA Science and Technology Conference*, 2007.
- [26] K. R. Britting, *Inertial Navigation Systems Analysis*, John Wiley & Sons, New York, NY, USA, 1971.
- [27] R. S. Bucy and K. D. Senne, "Digital synthesis of nonlinear filters," *Automatica*, no. 7, pp. 287–298, 1971.
- [28] R. M. Rogers, *Applied Mathematics in Integrated Navigation Systems*, American Institute of Aeronautics & Astronautics, Reston, Va, USA, 2000.
- [29] G. Zhang and L. Shen, "Rule-based expert system for selecting scene matching area," in *Intelligent Control and Automation*, vol. 344, pp. 546–553, Springer, Berlin, Germany, 2006.
- [30] G. Cao, X. Yang, and S. Chen, "Robust matching area selection for terrain matching using level set method," in *Proceedings of the 2nd International Conference on Image Analysis and Recognition (ICIAR '05)*, vol. 3656 of *Lecture Notes in Computer Science*, pp. 423–430, Toronto, Canada, September 2005.
- [31] E. L. Hall, D. L. Davies, and M. E. Casey, "The selection of critical subsets for signal, image, and scene matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, no. 4, pp. 313–322, 1980.