VOL. 2, NO. 11, November 2012 ISSN 2222-9833
ARPN Journal of Systems and Software
©2009-2012 AJSS Journal. All rights reserved

http://www.scientific-journals.org

Low-cost Vision Sensors and Integrated Systems for Unmanned Aerial Vehicle Navigation

Roberto Sabatini, Celia Bartel, Tesheen Shaid, Anish Kaharkar
Cranfield University – Department of Aerospace Engineering, Cranfield, Bedford MK43 0AL, UK

ABSTRACT

A novel low cost navigation system based on Vision Based Navigation (VBN) and other avionics sensors is presented, which is designed for small size Unmanned Aerial Vehicle (UAV) applications. The main objective of our research is to design a compact, light and relatively inexpensive system capable of providing the required navigation performance in all phases of flight of a small UAV, with a special focus on precision approach and landing, where Vision Based Navigation (VBN) techniques can be fully exploited in a multisensory integrated architecture. Various existing techniques for VBN are compared and the Appearance-based Navigation (ABN) approach is selected for implementation. Feature extraction and optical flow techniques are employed to estimate flight parameters such as roll angle, pitch angle, deviation from the runway and body rates. Additionally, we address the possible synergies between VBN, Global Navigation Satellite System (GNSS) and MEMS-IMU (Micro-Electromechanical System Inertial Measurement Unit) sensors and also the use of Aircraft Dynamics Models (ADMs) to provide additional information suitable to compensate for the shortcomings of VBN and MEMS-IMU sensors in high-dynamics attitude determination tasks. An Extended Kalman Filter (EKF) is developed to fuse the information provided by the different sensors and to provide estimates of position, velocity and attitude of the UAV platform in real-time. Two different integrated navigation system architectures are implemented. The first uses VBN at 20 Hz and GPS at 1 Hz to augment the MEMS-IMU running at 100 Hz. The second mode also includes the ADM (computations performed at 100 Hz) to provide augmentation of the attitude channel. Simulation of these two modes is performed in a significant portion of the AEROSONDE UAV operational flight envelope and performing a variety of representative manoeuvres (i.e., straight climb, level turning, turning descent and climb, straight descent, etc.). Simulation of the first integrated navigation system architecture (VBN/IMU/GPS) shows that the integrated system can reach position, velocity and attitude accuracies compatible with CAT-II precision approach requirements. Simulation of the second system architecture (VBN/IMU/GPS/ADM) also shows promising results since the achieved attitude accuracy is higher using the ADM/VBS/IMU than using VBS/IMU only. However, due to rapid divergence of the ADM virtual sensor, there is a need for frequent re-initialisation of the ADM data module, which is strongly dependent on the UAV flight dynamics and the specific manoeuvring transitions performed.

Keywords: *Unmanned Aerial Vehicle, Vision Based Navigation, MEMS Inertial Measurement Unit, GPS, Low-cost Navigation Sensors.*

1. INTRODUCTION

The use of Unmanned Aerial Vehicles (UAVs) in civil and military applications has much increased as these vehicles provide cost-effective and safe alternatives to manned flights in several operational scenarios. These robotic aircraft employ a variety of sensors, as well as multi-sensor data fusion algorithms, to provide autonomy to the platform in the accomplishment of mission- and safety-critical tasks. UAVs are characterized by higher manoeuvrability, reduced cost, longer endurance and less risk to human life compared to manned systems. Flight operations depend primarily upon accurate and continuous knowledge of vehicular position and attitude. In the case of an aircraft, this is required primarily to provide guidance information to the pilot. Similarly, in the case of a UAV, continuous and accurate position and attitude data are required to allow platform guidance and control.[1]

These tasks can be accomplished by a remote pilot and, frequently, implementing automatic flight modes. Technical requirements for air navigation systems primarily include accuracy, physical characteristics such as weight and volume, support requirements such as electrical power, and system integrity. One of the most important concepts is to use a multi-sensor integrated system to cope with the requirements of long/medium range navigation and landing. This would reduce cost, weight/volume and support requirements and, with the appropriate sensors and integration architecture, give increased accuracy and integrity of the overall system. The best candidates for such integration are indeed satellite navigation receivers and inertial sensors. In recent years, computer vision and Vision-Based Navigation (VBN) systems have started to be applied to UAVs. VBN can provide a self-contained autonomous navigation solution and can be used as an alternative (or an addition) to the traditional sensors (GPS, INS and integrated GPS/INS). The required information to perform autonomous navigation can be obtained from cameras which are compact and lightweight sensors.

[1] Corresponding author: r.sabatini@cranfield.ac.uk, phone +44 (0)1234 758290, www.cranfield.ac.uk

This is particularly attractive in UAV platforms, where weight and volume are tightly constrained. Sinopoli [1] used a model-based approach to develop a system which processes image sequences from visual sensors fused with readings from GPS/INS to update a coarse, inaccurate 3D model of the surroundings. Digital Elevation Maps (DEM) were used to build the 3D model of the environment. Occupancy Grid Mapping was used in this study, in which the maps were divided into cells. Each cell had a probability value of an obstacle being present associated with it. Using this 'risk map' and the images from the visual sensors, the UAV was able to update its stored virtual map. Shortest path optimisation techniques based on the Djikstra algorithm and Dynamic Programming were then used to perform obstacle avoidance and online computation of the shortest trajectory to the destination. Se [2] proposed a system which deals with vision-based SLAM using a trinocular stereo system. In this study, Scale-Invariant Feature Transform (SIFT) was used for tracking natural landmarks and to build the 3D maps. The algorithm built submaps from multiple frames which were then merged together. The SIFT features detected in the current frame were matched to the pre-built database map in order to obtain the location of the vehicle. With the development of deep space exploration, vision-based navigation has been used to provide an autonomous navigation system for rovers operating on the lunar surface and on Mars. The use of GPS on the Moon is not possible as its signals cannot reach the lunar surface. Sonar and magnetic sensors cannot be used due to the absence of an atmosphere and magnetic field, while dead reckoning sensors such as INS suffer from error accumulation. Stereo vision-based navigation has been used for navigating these space rovers because of its low power consumption and high reliability. The stereo vision system provides motion estimation for autonomous localisation techniques and environmental information for obstacle avoidance and DEM construction. A stereo vision-based navigation system was implemented on the 2004 twin Mars Exploration Rovers, Spirit and Opportunity. They were equipped with three types of stereo cameras which allowed them to navigate the Martian surface for long distances. A stereo vision-based navigation system for lunar rovers was proposed by Cui [3]. The system performed robust motion estimation and disparity estimation using stereo images. This enabled accurate lunar navigation, obstacle avoidance and DEM reconstruction. The position of the rover while descending was found using sequence images taken by a descent camera and the reading were integrated with measurements from a laser altimeter and DEM. Image processing was carried out using feature detection, tracking and stereo matching. Levenberg-Marquardt non-linear estimation was used for motion estimation and Weighted Zero Sum of Squared Differences gave the disparity estimation. Matsumoto [4] proposed a representation of the visual route taken by robots in appearance-based navigation. This approach was called the View-Sequenced Route Representation (VSRR) and was a sequence of images memorised in the recording run along the required route. The visual route connected the initial position and destination via a set of images. This visual route was used for localisation and guidance in the autonomous run. Pattern recognition was achieved by matching the features detected in the current view of the camera with the stored images. Criteria for image capture during the learning stage were given in the study. The visual route was learnt while the robot was manually guided along the required trajectory. A matching error between the previous stored image and current view was used to control the capture of the next key image. The current view was captured and saved as the next key image when a pre-set error threshold was exceeded. Localisation was carried out at the start of the autonomous run by comparing the current view with the saved visual route images. The key image with the greatest similarity to the current view represented the start of the visual route. The location of the robot depended purely on the key image used and no assumption was made of its location in 3D space. During the autonomous run, the matching error between the current view and key images was monitored in order to identify which image should be used for guidance. The robot was controlled so as to move from one image location to another and finally reach its destination. This 'teach-and-replay' approach was adopted by Courbon [5] [6], Chen [7] and Remazeilles [8]. In the case of [5] and [6], a single camera and natural landmarks were used to navigate a quadrotor UAV along the visual route. The key images were considered as waypoints to be followed in sensor space. Zero Normalised Cross Correlation was used for feature matching between the current view and the key images. A control system using the dynamic model of the UAV was developed. Its main task was to reduce the position error between the current view and key image to zero and to stabilize and control the UAV. Vision algorithms to measure the attitude of a UAV using the horizon and runway were presented by Xinhua [9] and Dusha [10]. The horizon is used by human pilots to control the pitch and roll of the aircraft while operating under visual flying rules. A similar concept is used by computer vision to provide an intuitive means of determining the attitude of an aircraft. This process is called Horizon-Based Attitude Estimation (HBAE). In [9], grayscale images were used for image processing. The horizon was assumed to be a straight line and appeared as an edge in the image. Texture energy method was used to detect it and this was used to compute the bank and pitch angle of the UAV. The position of the UAV with respect to the runway was found by computing the angles of the runway boundary lines. A Canny Edge detector was applied to part of the image below the horizon. The gradient of the edges was computed using the pixel coordinates which gave a rough indication of where the UAV was situated with respect to the runway. Dusha [10] used a similar approach to develop algorithms to compute the attitude and attitude rates. A Sobel edge detector was applied to each channel of the RGB image. The three channels were then combined and Hough transform was used to detect the horizon. In this research, it was assumed that the camera frame and the body frames were coincidental and equations were developed so as to calculate the pitch and roll angle. The angular rates of the UAV were derived using optical flow of the horizon. Optical flow gives us additional information of the states of the UAV and is dependent on the angular rates, velocity and the distance of the detected features. During this research, it was observed that the image processing frontend was susceptible to false detection of the horizon if any other strong edges were present in the image. Therefore, an Extended Kalman Filter (EKF) was implemented to filter out these incorrect results. The performance of the algorithms was tested via test flights with a small UAV and a Cessna 172. Results of the test flight with the UAV showed an error in the calculated pitch and roll with standard deviations of 0.42 and 0.71 degrees respectively. Moving forward from these results, in our research we designed and tested a new VBN sensor specifically tailored for apparoach/landing applications which, in addition to horizon detection and image-flow, also employed runway features extraction during the approach phase. Additionally, various candidates were considered for integration with the VBN sensor, including Global Navigation Satellite Systems (GNSS) and Micro Electro Mechanical Systems (MEMS) based Inertial Measurement Units (IMUs). MEMS-IMUs are low-cost and low-volume/weight sensors particularly well suited for small/medium size UAV applications. However, their integration represent a challenge, which need to be addressed either by finding improvements to the existing analytical methods or by developing novel algorithmic approaches that counterbalance the use of less accurate inertial sensors. In line with the above discussions, the main objective of our research is to develop a low-cost and low-weight/volume Navigation and Guidance System (NGS) based on VBN and other low-cost and low-weight/volume sensors, capable of providing the required level of performance in all flight phases of a small/medium size UAV, with a special focus on precision approach and landing (i.e., the most demanding and potentially safety-critical flight phase), where VBN techniques can be fully exploited in a multisensory integrated architecture. The NGS is composed by a Multisensor Integrated Navigation System (MINS) using an Extended Kalman Filter (EKF) and an existing controller that employs Fuzzy logic and Proportional-Integral-Differential (PID) technology.

2. VBN SENSOR DESIGN, DEVELOPMENT AND TEST

As discussed above, VBN techniques use optical sensors (visual or infrared cameras) to extract visual features from images which are then used for localization in the surrounding environment. Cameras have evolved as attractive sensors as they help design economically viable systems with simpler hardware and software components. Computer vision has played an important role in the development of UAVs [11]. Considerable work has been made over the past decade in the area of vision-based techniques for navigation and control [9]. UAV vision-based systems have been developed for various applications ranging from autonomous landing to obstacle avoidance. Other applications looked into the possible augmentation INS and GPS/INS by using VBN measurements [12]. As discussed above, several VBN sensors and techniques have been developed. However, the vast majority of VBN sensor schemes fall into one of the following two categories [13]: Model-based Approach (MBA) and Appearance-based Approach (ABA). MBA uses feature tracking in images and create a 3D- model of the workspace in which robots or UAV operates [14]. The 3D maps are created in an offline process using a priori information of the environment. Localisation is carried out using feature matching between the current view of the camera and the stored 3D model. The orientation of the robot is found from 3D-2D correspondence. MBA has been extensively researched in the past and is the most common technique currently implemented for vision-based navigation. However, the accuracy of this method is dependent on the features used for tracking, robustness of the feature descriptors and the algorithms used for matching and reconstruction. The reconstruction in turn relies on proper camera calibration and sensor noise. Knowledge of surroundings so as to develop the 3D models is also required prior to implementation which may not be case in most situations. ABA algorithms eliminate the need for a metric model as they work directly in the sensor space. This approach utilizes the appearance of the whole scene in the image, contrary to model-based approach which uses distinct objects such as landmarks or edges [4]. The environment is represented in the form of key images taken at various locations using the visual sensors. This continuous set of images describes the path to be followed by the robot. The images are captured while manually guiding the robot through the workspace. In this approach, localisation is carried out by finding the key image with the most similarity to the current view. The robot is controlled by either coding the action required to move from one key image to another or by a more robust approach using visual servoing [8] [15]. The ABA approach is relatively new and has gained active interest. The modelling of the surrounding using a set of key images is more straightforward to implement compared to 3D modelling. A major drawback of this method is its limited applicability. The robot assumes that the key image database for a particular workspace is already stored in its memory. Therefore, the key images need to be recaptured each time the robot moves to a new workspace. It is limited to work in the explored regions which have been visualised during the learning stage [16]. The ABA approach has a disadvantage in requiring a large amount of memory to store the images and is computationally more costly than MBA. However, due to improvements in computer technology, this technique has become a viable solution in many application areas. We selected the ABA approach for the design of our VBN sensor system.

2.1 Learning Stage

The first step required for appearance based navigation is the learning stage. During this stage, a video is recorded using the on-board camera while guiding the aircraft manually during the landing phase. The recorded video is composed of a series of frames which form the visual route for landing. This series of frames is essentially a set of images connecting the initial and target location images. The key frames are first sampled and the selected images are stored in the memory to be used for guidance during autonomous landing of the aircraft. During the learning stage, the UAV is flown manually meeting the Required Navigation Performance (RNP) requirements of precision approach and landing. If available, Instrument Landing System (ILS) can also be used for guidance. It should be noted that the visual route captured while landing on a runway, can only be used for that particular runway. If the UAV needs to land at multiple runways according to its mission, the visual route for all the runways is required to be stored in the memory. The following two methods can be employed for image capture during the learning stage.

- **Method 1:** Frames are captured from the video input at fixed time intervals. The key frames are selected manually in this case.

- **Method 2:** Frames are captured using a matching difference threshold [4]. This matching difference threshold is defined in number of pixels and can be obtained by tracking the features in the current view and the previously stored key image. The key images can then be selected based on the threshold and stored in the memory.

The algorithm starts by taking an image at the starting point. Let this image be M_i captured at location i. As the aircraft moves forward, the difference between the current view (V) and the image M_i increases. This difference keeps increasing until it reaches the set threshold value (x). At this point, a new image M_{i+1} is taken (replacing the previous image M_i) and the process is repeated until the aircraft reaches its destination. The learning stage process is summarised in Fig. 1(a).

2.2 Localisation

Localisation is a process which determines the current location of the aircraft at the start of autonomous run. This process identifies the key image which is the closest match to the current view. The current view of the aircraft is compared with a certain number of images, preferably the ones at the start of the visual route. The key image with the least matching difference is considered to be the start of the visual route to be followed by the aircraft. At the start of the autonomous run, the aircraft is approximately at the starting position of the visual route. The current view, captured from the on-board camera is compared with a set of images (stored previously in the memory during the learning stage) in order to find the location of aircraft with respect to the visual route. The key image with the least matching difference with the current view is considered to be the location of the aircraft and marks the start of the visual route to be followed. The process is summarised in Fig. 1(b).

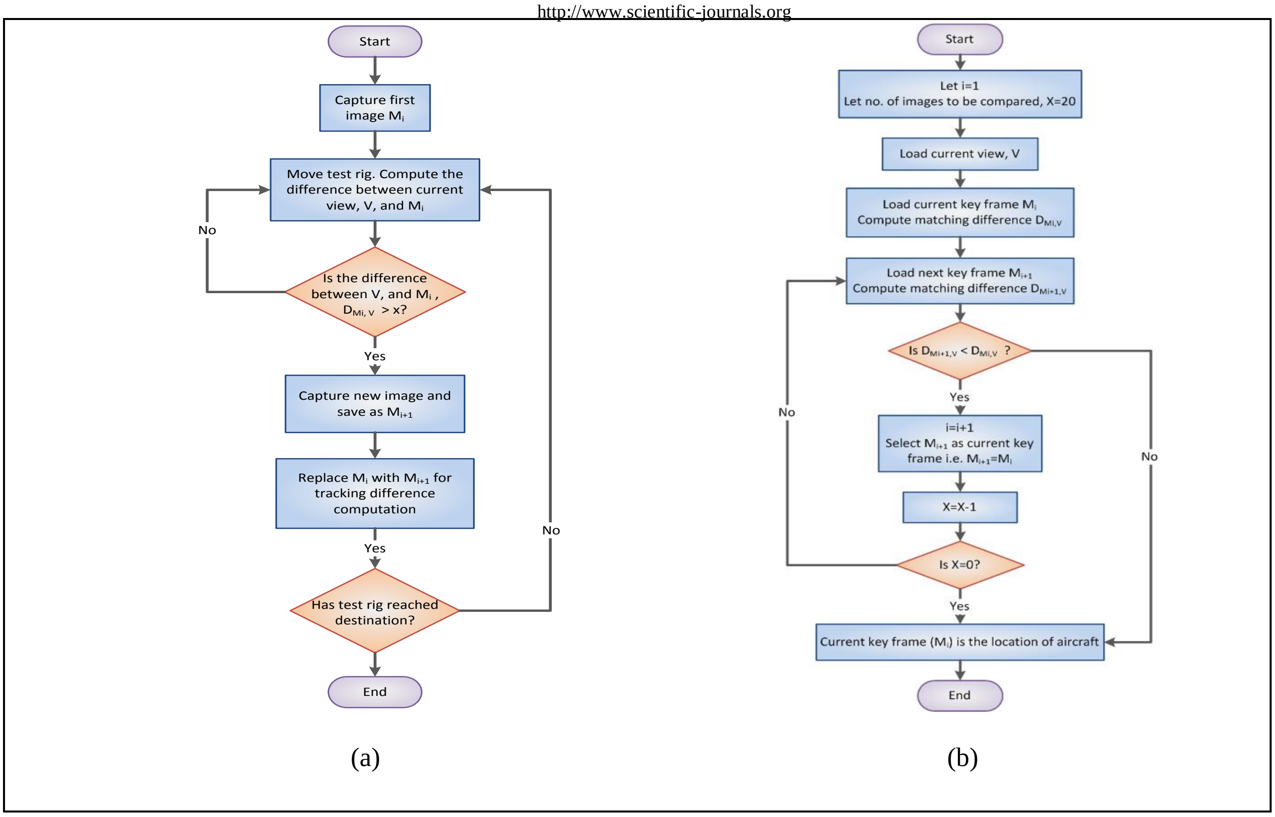
Start
Capture first image M_i
Move test rig. Compute the difference between current view, V, and M_i
No
Is the difference between V, and M_i, $D_{Mi,V}$ > x?
Yes
Capture new image and save as M_{i+1}
Replace M_i with M_{i+1} for tracking difference computation
Yes
No
Has test rig reached destination?
End
(a)
Start
Let i=1
Let no. of images to be compared, X=20
Load current view, V
Load current key frame M_i
Compute matching difference $D_{Mi,V}$
Load next key frame M_{i+1}
Compute matching difference $D_{Mi+1,V}$
Is $D_{Mi+1,V} < D_{Mi,V}$?
Yes
No
i=i+1
Select M_{i+1} as current key frame i.e. M_{i+1}=M_i
No
X=X-1
Is X=0?
Yes
Current key frame (M_i) is the location of aircraft
End
(b)

Figure 1: Image capture (a) and localisation (b).

In this example, the number of images to be compared (X) is taken as 20. First, the algorithm loads the current view (V) and the first key frame (M_i). Then the difference between the current view and the current key frame is computed. The algorithm then loads the next key frame M_{i+1} and again computes the difference with the current view. If this difference is less than the previous difference, M_{i+1} replaces M_i, and the process is repeated again. Otherwise, M_i is considered as the current location of the aircraft.

2.3 Autonomous Run

During the autonomous run phase, the aircraft follows the visual route (previously stored in memory during the learning stage) from the image identified as the current location of the aircraft during localisation. The set of key images stored as the visual route can be considered as the target waypoints for the aircraft in sensor space. The current view is compared to the key images so as to perform visual servoing. The approach followed to identify the key image to be used for visual servoing, is describes as follows. Let M_j be the current key frame, i.e. image with the least matching difference with the current view. During the autonomous run, the current key image and the next key image (M_{j+1}) are loaded. The matching differences of the current view V with M_j and M_{j+1} (which are $D_{Mj,V}$ and $D_{Mj+1,V}$ respectively) are tracked. When the matching difference $D_{Mj,V}$ exceeds $D_{Mj+1,V}$, M_{j+1} is taken as the current key image replacing M_j and the next key image is loaded as M_{j+1}. This same process keeps repeating until the aircraft reaches its destination, that is the final key frame. Fig. 2(a) summarises the process of autonomous run in the form of a flow chart while the change in matching difference for different key frames during autonomous run is presented in Fig. 2(b).

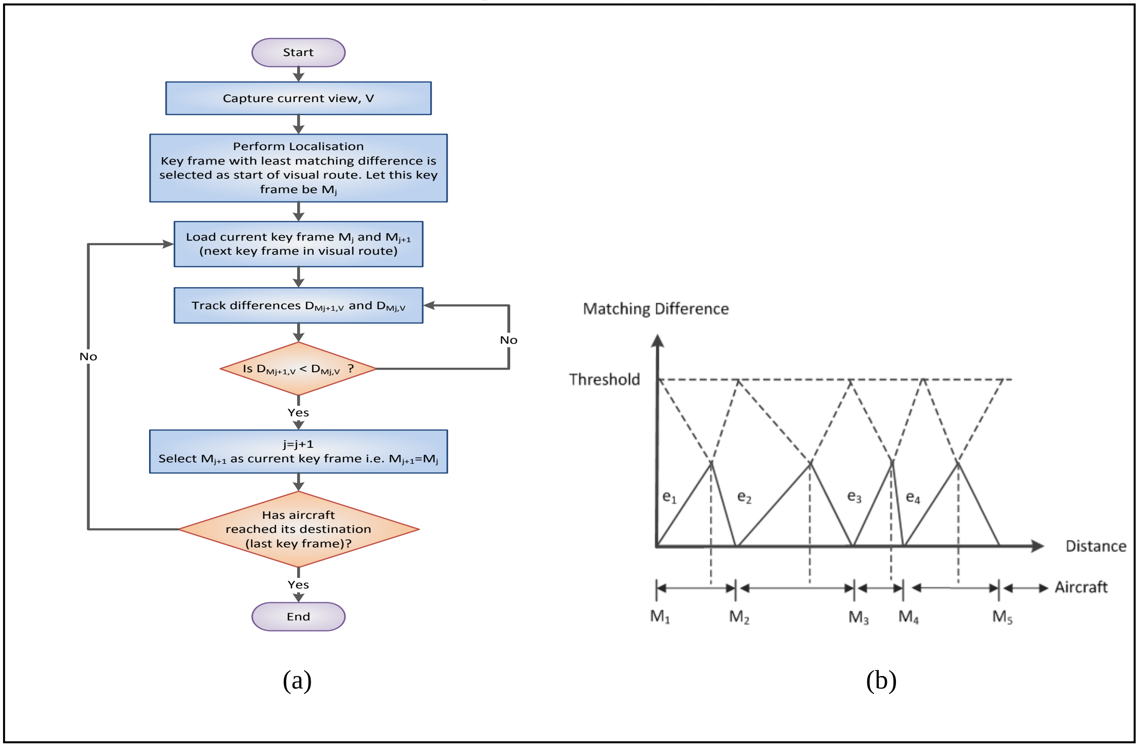
Start
Capture current view, V
Perform Localisation
Key frame with least matching difference is selected as start of visual route. Let this key frame be M_j
Load current key frame M_j and M_{j+1} (next key frame in visual route)
Track differences $D_{Mj+1,V}$ and $D_{Mj,V}$
Is $D_{Mj+1,V} < D_{Mj,V}$?
No
No
Yes
j=j+1
Select M_{j+1} as current key frame i.e. $M_{j+1}=M_j$
Has aircraft reached its destination (last key frame)?
Yes
End
(a)
Matching Difference
Threshold
e_1
e_2
e_3
e_4
Distance
Aircraft
M_1
M_2
M_3
M_4
M_5
(b)

Figure 2: Autonomous run (a) and matching difference/key-frame selection (b).

The proposed vision based navigation process is depicted in Fig. 3. The key frames represent the visual route the aircraft requires to follow.

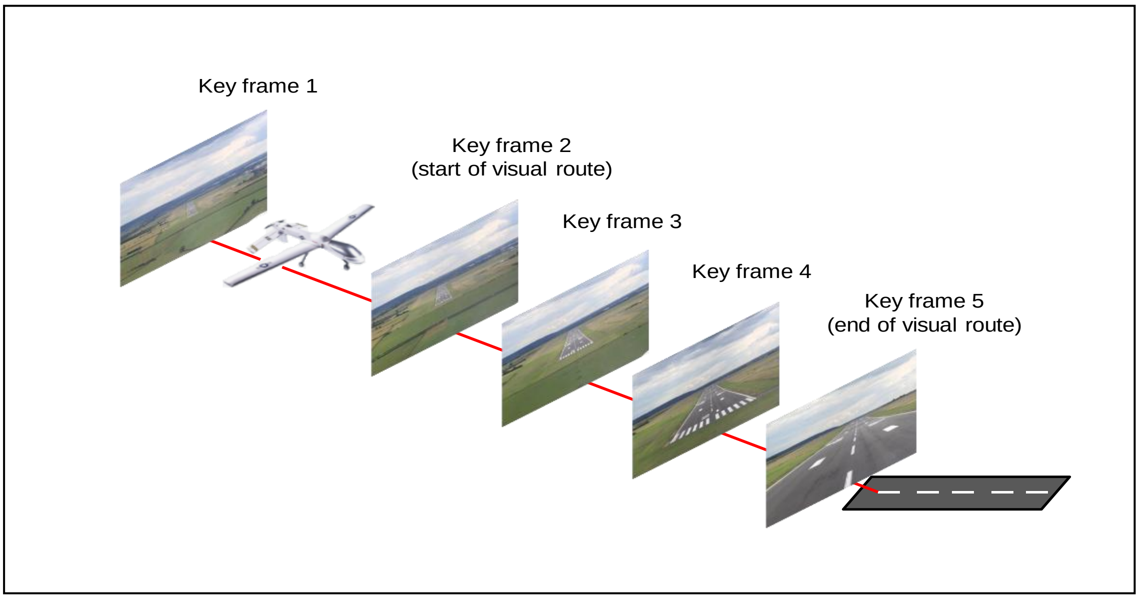

Figure 3: VBN process.

The figure shows that the key frame 2 is identified as the starting point of the visual route during the localisation process. The onboard computer tracks the matching difference between current view and the second and third key frames until the difference for key frame 2 and the current view exceeds the difference of key frame 3 and the current view. At this stage, key image 3 is used to control the UAV and the matching differences between key frames 3, 4 and the current view are monitored. This process is repeated until the UAV reaches its destination. To capture the outside view, a monochrome Flea camera from Point Grey Research was used. The main specification of the camera and lenses are listed in Table 1.

Table 1: Point Grey Flea and Lenses Specifications.

Sensor type	Sony ICX204AQ/AL 1/3" CCD sensor Scan type : Progressive
Resolution	1024 x 768 BW
Format	8-bit or 16-bit, 12-bit AtoD
Pixel Size	4.65µm x 4.65µm
Frame Rates	1.875, 3.75, 7.5, 15, 30fps
Video output signal	8 bits per pixel / 12 bits per pixel digital data
Interfaces	6-pin IEEE-1394 for camera control and video data transmission 4 general purpose digital input/output pins
Voltage Requirements	8-32V
Power Consumption	< 3W
Gain	Automatic/Manual modes at 0.035dB resolution (0 to 24dB)
Shutter	Automatic/Manual/Extended Shutter modes (20 µs to 66ms @ 15Hz)
Trigger Modes	DCAM v1.31 Trigger Modes 0, 1 and 3
SNR	50dB or better at minimum gain
Camera Dimensions (no lenses)	30mm x 31mm x 29mm
Mass	60g without optics
Operating Temperature	0° to 45°C
Focal length	3.5- 8.0 mm
Max CCD Format	1/3"
Aperture	F1.4 – 16 (closed) - Manual control
Maximum Field of View (FOV)	Horizontal: 77.6° / Vertical: 57.6°
Min working distance	0.4m
Lenses Dimensions (mm)	34.0 diameter x 43.5 length

This camera was also used in a previous study on stereo vision [17] and was selected for this project. The Flea camera is a pinhole charged coupled device (CCD) camera with a maximum image resolution of 1024 x 768 pixels. It is capable of recording videos at a maximum rate of 30 fps. An IEEE 1394 connection was used to interface the camera and computer with a data transfer speed of 400 Mbps. A fully manual lens which allows manual focus and zoom was fitted to the camera. A UV filter was employed to protect the lens and to prevent haziness due to ultraviolet light.

2.4 Image Processing Module

The Image Processing Module (IPM) of the VBN system detects horizon and runway centreline from the images and computes the aircraft attitude, body rates and deviation from the runway centreline. Fig. 4a shows the functional architecture of the IPM.

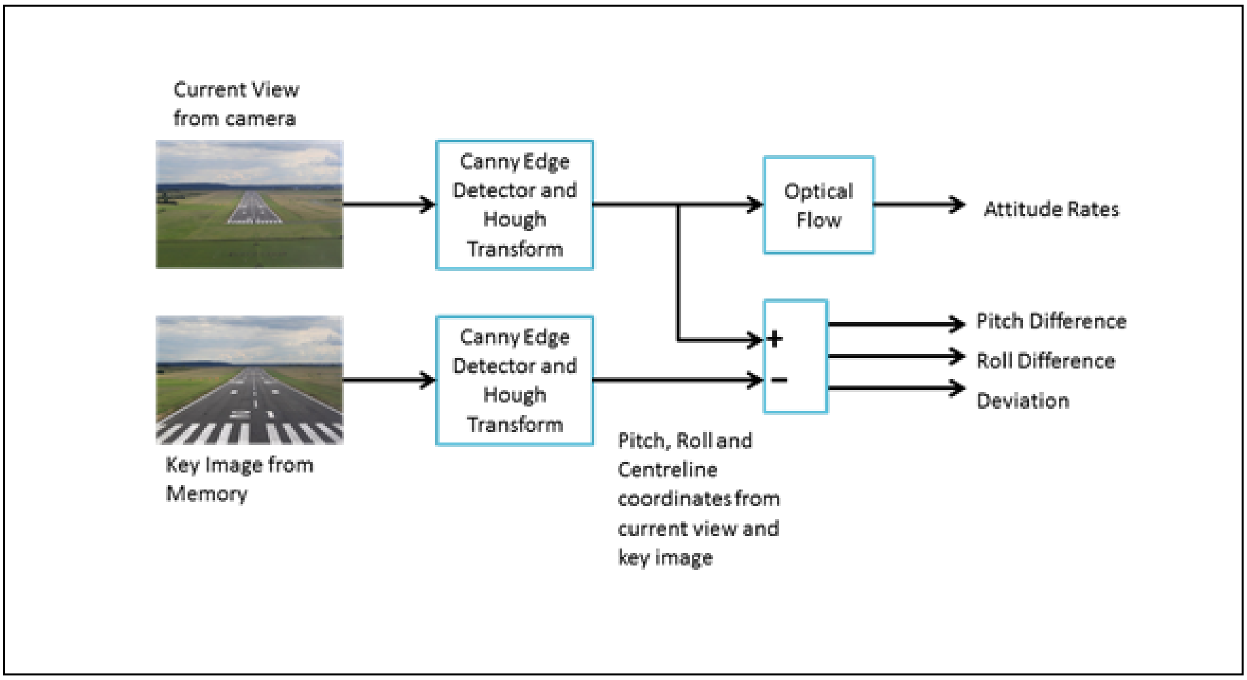
Current View from camera
Canny Edge Detector and Hough Transform
Optical Flow
Attitude Rates
Canny Edge Detector and Hough Transform
+
–
Pitch Difference
Roll Difference
Deviation
Key Image from Memory
Pitch, Roll and Centreline coordinates from current view and key image

Figure 4a: Functional Architecture of the IPM.

The detailed processing performed by the IPM is illustrated in Fig. 5. As a first step, the size of the image is reduced from 1024x768 pixels to 512x384 pixels. After some trials, it was found that this size reduction speeds up the processing without significantly affecting the features detection process. The features such as the horizon and the runway centreline are extracted from the images for attitude computation. The horizon is detected in the image by using Canny edge detector while the runway centreline is identified with the help of Hough Transform. The features are extracted from both, the current view (image received from the on-board camera) and the current key frame. The roll and pitch are computed from the detected horizon while the runway centreline in used to compute the deviation of aircraft from the runway centreline. Then the roll and pitch difference are computed between the current view and the current key frame. Optical flow is determined for all the points on the detected horizon line in the images. The aircraft body rates are then computed based on the optical flow values. The image processing module provides the aircraft attitude, body rates, pitch and roll differences between current view and key frame, and deviation from the runway centreline. The attitude of the aircraft is computed based on the detected horizon and the runway. The algorithm calculates the pitch and roll of the aircraft using the horizon information while aircraft deviation from the runway centreline is computed using the location of runway centreline in the current image.

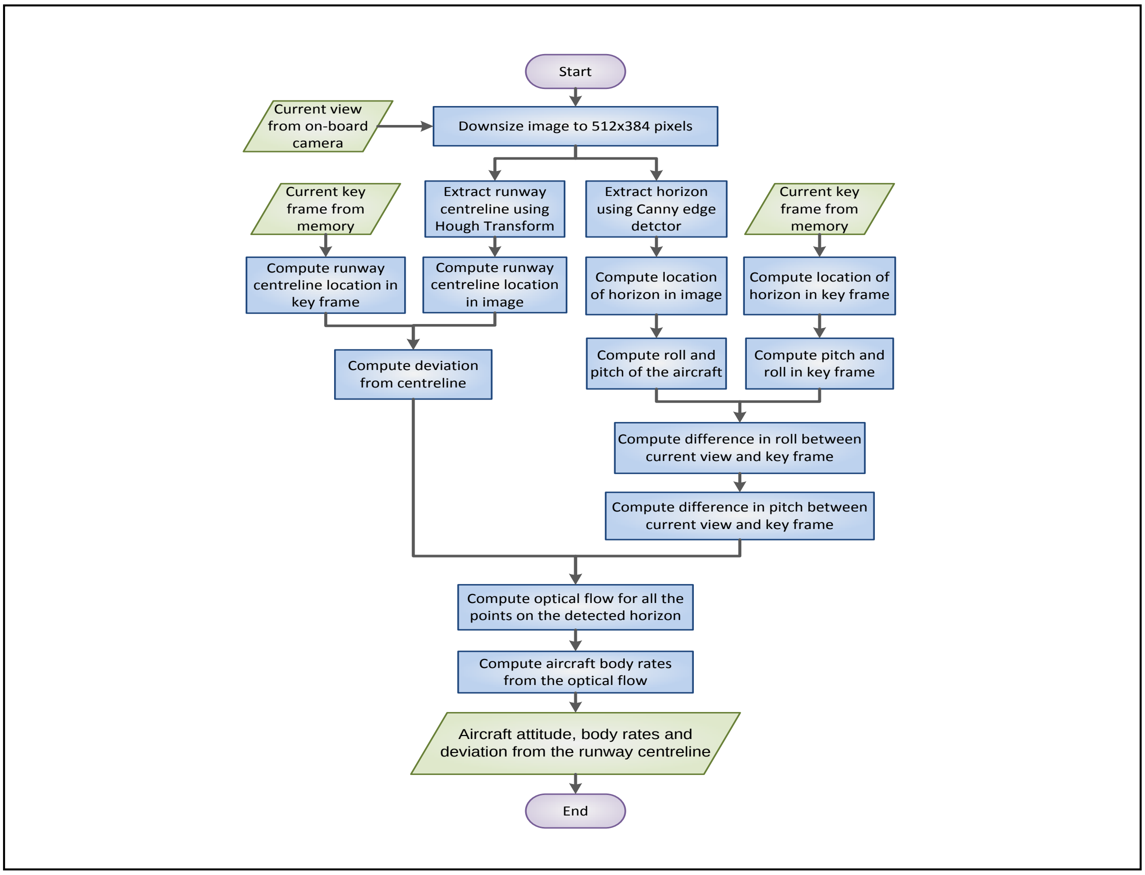
Start
Current view from on-board camera
Downsize image to 512x384 pixels
Current key frame from memory
Extract runway centreline using Hough Transform
Extract horizon using Canny edge detector
Current key frame from memory
Compute runway centreline location in key frame
Compute runway centreline location in image
Compute location of horizon in image
Compute location of horizon in key frame
Compute deviation from centreline
Compute roll and pitch of the aircraft
Compute pitch and roll in key frame
Compute difference in roll between current view and key frame
Compute difference in pitch between current view and key frame
Compute optical flow for all the points on the detected horizon
Compute aircraft body rates from the optical flow
Aircraft attitude, body rates and deviation from the runway centreline
End

Figure 4b: Image processing module flowchart.

Fig. 5 shows the relationship between the body (aircraft) frame (O_b, X_b,Y_b, Z_b), camera frame (O_c, X_c,Y_c, Z_c) and Earth frame coordinates (O_w, X_w, Y_w, Z_w).

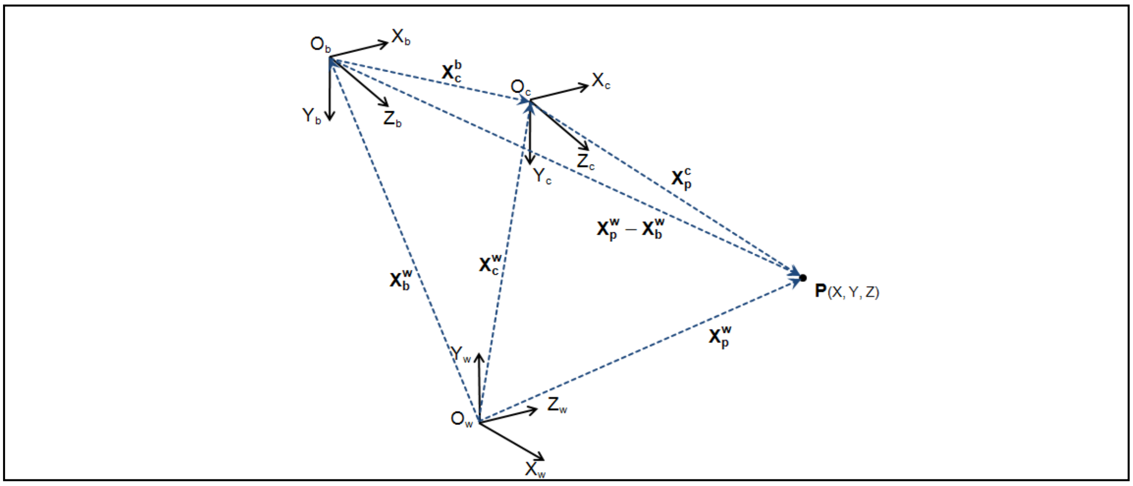

Figure 5: Coordinate system.

The position of a 3D space point P in Earth coordinates is represented by a vector X_p^w with components x_p, y_p and z_p in the Earth frame. The position of aircraft centre with respect to the Earth coordinates is represented by the vector X_b^w with components x_b, y_b and z_b in the Earth frame. The vector X_p^c represents the position of the point P with respect to the camera frame with components x_{cp}, y_{cp} and z_{cp} in the camera frame. The position of centre of camera lens with respect to the body frame is represented by the vector X_c^b. The vector X_c^w represents the position of lens centre with respect to the ground frame with components x_c, y_c and z_c in the ground frame. The position of point P with respect to body frame with components in the Earth frame can be computed as X_p^w-X_b^w. The transformation matrix from Earth frame to body frame C_w^b can be obtained in terms of the yaw ψ, pitch θ, and roll angle ϕ as:

$$C_w^b = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix} \tag{1}$$

The position of point P with respect to the aircraft's body with components in the body frame can be obtained as:

$$X_p^b = C_w^b\left(X_p^w - X_b^w\right) \tag{2}$$

The position of point P with respect to the camera frame can also be found in a similar way as:

$$X_p^c = C_b^c\left(X_p^b - X_c^b\right) = C_b^c C_w^b\left(X_p^w - X_b^w\right) - C_b^c X_c^b \tag{3}$$

where C_b^c is the constant transformation matrix from the body frame to the camera frame. With the assumption that the camera is fixed with respect to the body and the angle from the camera optical axis to the longitudinal axis of the aircraft is fixed value, $-C_b^c X_c^b$ is a known constant vector with components k_x, k_y and k_z in the camera frame. In this case, the velocity and rotation rate of aircraft are the same as those of the camera. Thus, the position and attitude of the aircraft can be easily computed according to those of the camera as:

$$X_b^w = X_c^w - C_c^w C_b^c X_c^b = X_c^w - C_c^w\left[k_x, k_y, k_z\right]^T \tag{4}$$

$$\phi = \phi_c \tag{5}$$

$$\theta = \theta_c - \theta_0 \tag{6}$$

$$\psi = \psi_c \tag{7}$$

Where ϕ_c is the roll, θ_c is the pitch, ψ_c is the yaw and θ_0 is the angle of incidence of the camera. The transformation matrix from camera frame to the ground frame, represented by C_c^w, is obtained from:

$$C_w^b = \begin{bmatrix} \cos\theta_c\cos\psi_c & -\cos\phi_c\sin\psi_c + \sin\phi_c\sin\theta_c\cos\psi_c & \sin\phi_c\sin\psi_c + \cos\phi_c\sin\theta_c\cos\psi_c \\ \cos\theta_c\sin\psi_c & \cos\phi_c\cos\psi_c + \sin\phi_c\sin\theta_c\sin\psi_c & -\sin\phi_c\cos\psi_c + \cos\phi_c\sin\theta_c\sin\psi_c \\ -\sin\theta_c & \sin\phi_c\cos\theta_c & \cos\phi_c\cos\theta_c \end{bmatrix} \tag{8}$$

$$C_w^c = \left[C_c^w\right]^T \tag{9}$$

Eq. (9) represents the transformation matrix from the Earth frame to the camera frame coordinates [2]. From now onwards, only the state estimates of the camera are considered. The position of 3D point P with respect to camera frame is given by:

$$X_p^c = C_w^c\left(X_p^w - X_c^w\right) \tag{10}$$

with components in the camera frame given by:

$$\begin{cases} X = (-\cos\phi_c\sin\psi_c + \sin\phi_c\sin\theta_c\cos\psi_c)(x_p - x_c) + (\sin\phi_c\cos\theta_c)(z_p - z_c) \\ \qquad +(\cos\phi_c\cos\psi_c + \sin\phi_c\sin\theta_c\sin\psi_c)(y_p - y_c) \\ Y = (\sin\phi_c\sin\psi_c + \cos\phi_c\sin\theta_c\cos\psi_c)(x_p - x_c) + (\cos\phi_c\cos\theta_c)(z_p - z_c) \\ \qquad +(-\sin\phi_c\cos\psi_c + \cos\phi_c\sin\theta_c\sin\psi_c)(y_p - y_c) \\ Z = (\cos\theta_c\cos\psi_c)(x_p - x_c) + (\cos\theta_c\sin\psi_c)(z_p - z_c) + (-\sin\theta_c)(y_p - y_c) \end{cases} \tag{11}$$

Then, the coordinates (u, v) of P in the image plane can be obtained from:

$$u = \frac{fX}{Z} \tag{12}$$

$$v = \frac{fY}{Z} \tag{13}$$

Using the coordinate previously defined, the point P is assumed to be located on the detected horizon line. As the Earth's surface is approximated by a plane [39], a normal vector to the plane, n_w can be described as:

$$n_w = [0 \quad 1 \quad 0]^T \tag{14}$$

If the horizon line is described by a point X_p^w and a direction vector l_w tangential to the line of horizon visible to the image plane, then:

$$X_p^w = [x \quad 0 \quad d]^T \tag{15}$$

$$l_w = [1 \quad 0 \quad 0]^T \tag{16}$$

where x is an arbitrary point along x-axis and d is the distance to horizon along z-axis. If the camera is assumed to be placed directly above the origin of the ground frame, the position of camera X_c^w can be described as:

$$X_c^w = [0 \quad h \quad 0]^T \tag{17}$$

Then, a point on horizon may be expressed as:

$$X_p^w = X_p^c + X_c^w \tag{18}$$

The horizon projection on the image plane can be described by the point **p** and a direction vector **m** as:

$$m = \begin{bmatrix} m_x & m_y & 0 \end{bmatrix}^T \tag{19}$$

$$p = [u \quad v \quad f]^T \tag{20}$$

where (m_y/m_x) gives the gradient of the horizon line. As the position of the horizon X_p^w lies on the surface of the ground plane, therefore:

$$n_w \cdot X_p^w = 0 \tag{21}$$

Substituting X_p^w gives:

$$n_w \cdot (X_p^c + X_c^w) = 0 \tag{22}$$

The direction vector of the horizon line l_w lies on the plane and is therefore orthogonal to the normal vector. Therefore:

$$n_w \cdot l_w = 0 \tag{23}$$

Equations (22) and (23) are in form known as line plane correspondence problem as proposed by [38]. Recalling the equations for a projective perspective camera:

$$\begin{bmatrix} u \\ v \\ f \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{24}$$

Substituting Eq. (24) into Eq. (22), roll angle can be derived as [6]:

$$\phi = \tan^{-1}\left(\frac{-m_y}{m_x}\right) \tag{25}$$

which is an intuitive result that roll angle is dependent on the gradient of the horizon line on the image plane. Similarly, it can be shown that substituting Eq. (20) into Eq. (22), the pitch angle can be derived as [6]:

$$\theta = \tan^{-1}\left(\pm\frac{hf + ud\sin\phi + vd\cos\phi}{df - uh\sin\phi - vh\cos\phi}\right) \tag{26}$$

If the distance to the horizon is much greater than the height of the aircraft (i.e., $d \gg h$), the expression for pitch reduces to the following:

$$\theta = \tan^{-1}\left(\pm\frac{u\sin\phi + v\cos\phi}{f}\right) \tag{27}$$

which shows that pitch is dependent on roll angle, focal length and the position of horizon in the image plane. Optical flow depends on the velocity of the platform, angular rates of the platform and the distance to any features observed [6]. Differentiating Eq. (24) we obtain:

$$\dot{u} = \frac{f(\dot{X}Z - \dot{Z}X)}{Z^2} = \frac{f}{Z}\left(\dot{X} - \frac{X}{Z}\dot{Z}\right) \tag{28}$$

$$\dot{v} = \frac{f(\dot{Y}Z - \dot{Z}Y)}{Z^2} = \frac{f}{Z}\left(\dot{Y} - \frac{Y}{Z}\dot{Z}\right) \tag{29}$$

Substituting Eq. (28) into the time derivative of Eq. (18) yields the classical optical flow equations [6]:

$$\begin{bmatrix}\dot{u}\\ \dot{v}\end{bmatrix} = \frac{f}{Z}\begin{bmatrix}1 & 0 & \frac{-u}{f}\\ 0 & 1 & \frac{-v}{f}\end{bmatrix}\begin{bmatrix}\dot{r}_x^w\\ \dot{r}_y^w\\ \dot{r}_z^w\end{bmatrix} + \begin{bmatrix}\frac{uv}{f} & -\left(f + \frac{u^2}{f}\right) & v\\ \left(f + \frac{v^2}{f}\right) & -\frac{uv}{f} & -u\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \tag{30}$$

If the observed point lies on the horizon, then Z will be large and the translational component will be negligible. In this case, Eq. (30) reduces to:

$$\begin{bmatrix}\dot{u}\\ \dot{v}\end{bmatrix} = \begin{bmatrix}\frac{uv}{f} & -(f + \frac{u^2}{f}) & v\\ (f + \frac{v^2}{f}) & -\frac{uv}{f} & -u\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \tag{31}$$

To minimize the effect of errors, a Kalman filter is employed. The state vector consists of the roll angle, pitch angle and body rates of the aircraft. It is assumed that the motion model of the aircraft is disturbed by uncorrelated zero-mean Gaussian noise.

$$x(k+1) = f\big(k, x(k)\big) + \eta(k) \tag{32}$$

$$z(k) = h\big(k, x(k)\big) + w(k) \tag{33}$$

Where $x(k)$ is the state vector of the aircraft while $\eta(k)$ is the uncorrelated zero-mean Gaussian random vector with diagonal covariance matrix $Q(k)$. The measurement vector at time k is represented by $z(k)$ and $w(k)$ is the zero-mean Gaussian noise vector with a diagonal matrix $R(k)$. If the body rates are assumed to be approximately constant during the sampling interval Δt and first order integration is applied, then the state transition equations are as follows:

$$\begin{bmatrix} \phi(k+1) \\ \theta(k+1) \\ \omega_x(k+1) \\ \omega_y(k+1) \\ \omega_z(k+1) \end{bmatrix} = \begin{bmatrix} \phi(k) + \Delta t(\dot{\phi}(k)) \\ \theta(k) + \Delta t(\dot{\theta}(k)) \\ \omega_x(k) \\ \omega_y(k) \\ \omega_z(k) \end{bmatrix} + \begin{bmatrix} \eta_\phi(k) \\ \eta_\theta(k) \\ \eta_{\omega_x}(k) \\ \eta_{\omega_y}(k) \\ \eta_{\omega_z}(k) \end{bmatrix} \tag{34}$$

where

$$\dot{\phi}(k) = (\omega_x(k)\sin\big(\phi(k)\big) + \omega_y(k)\cos\big(\phi(k)\big)\tan\big(\phi(k)\big) + \omega_z(k) \tag{35}$$

$$\dot{\theta}(k) = \omega_x(k)\cos\big(\phi(k)\big) - \omega_y(k)\sin\big(\phi(k)\big) \tag{36}$$

The measurement equations are comprised of direct observations of the pitch and the roll from the horizon and i optical flow observations on the detected horizon line. Therefore, the length of the measurement vector $z(k)$ is $2(i+1)$. The relation of measurement vector and the states is represented by following linear equations:

$$\begin{bmatrix} \phi(k) \\ \theta(k) \\ \dot{u}_1(k) \\ \dot{v}_1(k) \\ \dot{u}_2(k) \\ \dot{v}_2(k) \\ \cdots \\ \dot{u}_i(k) \\ \dot{v}_i(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{u_1 v_1}{f} & -\left(f + \frac{{u_1}^2}{f}\right) & v_1 \\ 0 & 0 & \left(f + \frac{{v_1}^2}{f}\right) & \frac{-u_1 v_1}{f} & -u_1 \\ 0 & 0 & \frac{u_2 v_2}{f} & -\left(f + \frac{{u_2}^2}{f}\right) & v_2 \\ 0 & 0 & \left(f + \frac{{v_2}^2}{f}\right) & \frac{-u_2 v_2}{f} & -u_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \frac{u_i v_i}{f} & -\left(f + \frac{{u_i}^2}{f}\right) & v_i \\ 0 & 0 & \left(f + \frac{{v_i}^2}{f}\right) & \frac{-u_i v_i}{f} & -u_i \end{bmatrix} \begin{bmatrix} \phi(k) \\ \theta(k) \\ \omega_x(k) \\ \omega_y(k) \\ \omega_z(k) \end{bmatrix} \tag{37}$$

2.5 VBN Sensor Performance

Based on various laboratory, ground and flight test activities with small aircraft and UAV platforms, the performance of the VBN sensor were evaluated. Fig. 6 shows a sample image used for testing the VBN sensor algorithms and the results of the corresponding horizon detection process for attitude estimation purposes.

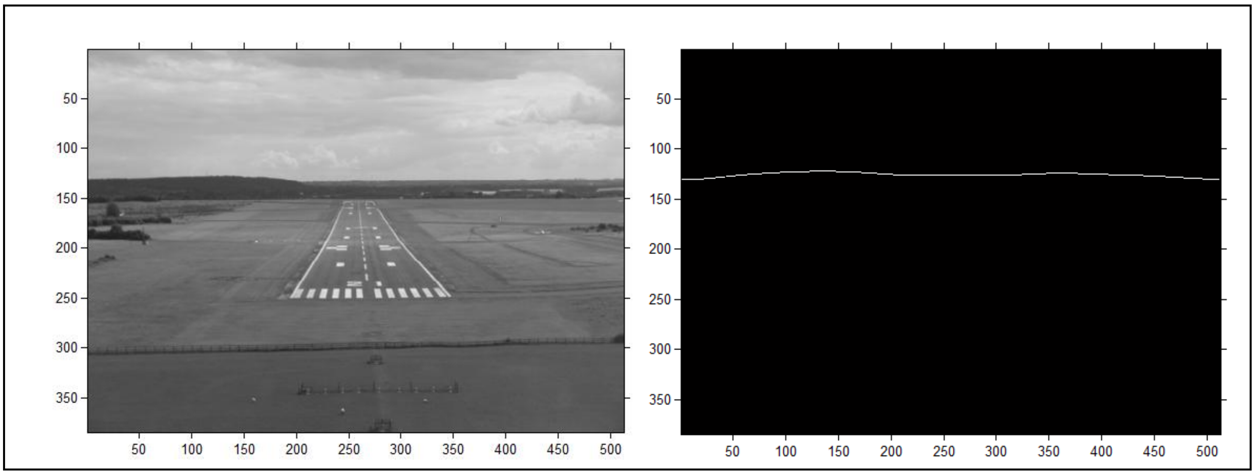

Figure 6: Horizon detected from the test image (landing phase).

The algorithm detects the horizon and the runway centreline from the images. The horizon is detected in the image by using Canny edge detector with a threshold of 0.9 and standard deviation of 50. in this experiment, the values of the threshold and the standard deviation were selected by hit-and-trial method. The resulting image after applying the Canny edge detector is a binary image. The algorithm assigns value '1' to the pixels detected as horizon while the rest of the pixels in the image are assigned value '0'. From this test image, the computed roll angle is 1.26° and the pitch angle is -10.17°. To detect the runway in the image, kernel filter and Hough Transform are employed. The runway detected from the same test image is shown in Fig. 7. For this image, the location of the runway centreline was computed in pixels as 261. The features were extracted from both the current view (image received from the on-board camera) and the current key frame. After the pitch, roll and centreline values were determined, the roll/pitch differences and the deviation from centreline are computed between the current view and the current key frame.

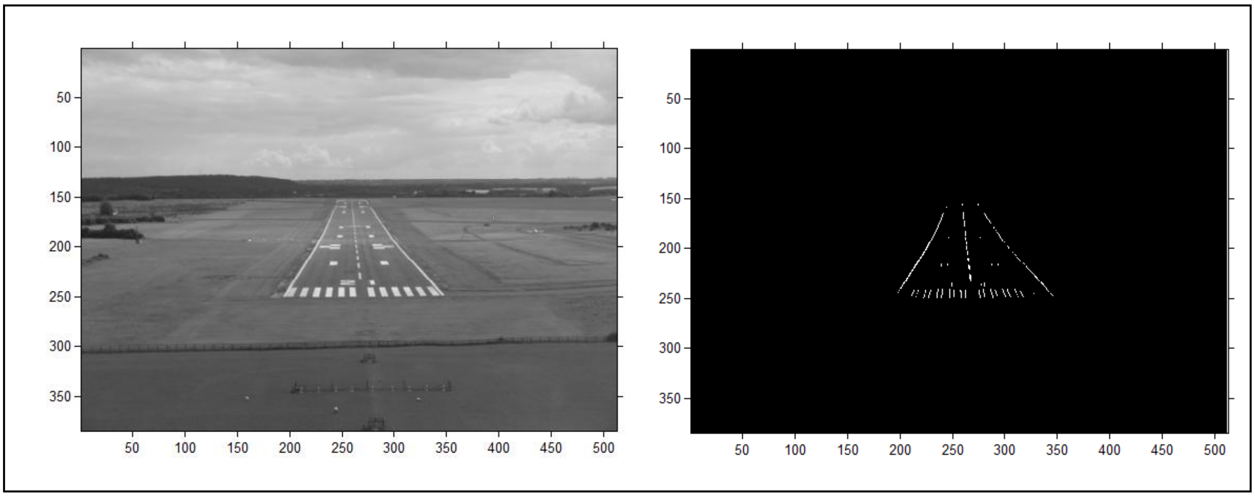

Figure 7: Runway detected in the test image.

The algorithm also computes the optical flow for all the points on the detected horizon line in the images. The optical flow is determined based on the displacement of points in two consecutive frames of a video. The algorithm takes two consecutive frames at a time and determines the motion for each point on the horizon. These optical flow values are used to compute the body rates of the aircraft. An example of the optical flow calculation is shown in Fig. 8, where the original image (from the camera) is shown on the left and the image on the right shows the optical flow vectors (in red) computed for the detected horizon line. The vectors are magnified by a factor of 20. Since the vectors on the right half of the horizon line are pointing upwards and the vectors on the left half are pointing downwards, the aircraft is performing roll motion (clockwise direction).

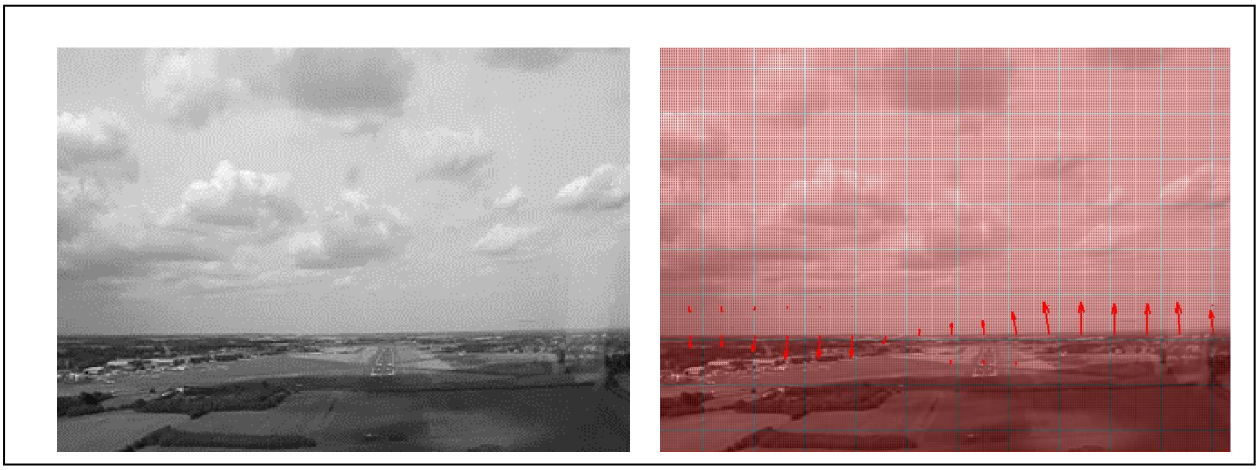

Figure 8: Received image from camera (left) and optical flow computed for the detected horizon in the image (right).

The real-time performance of the IPM algorithms were evaluated using a combination of experimental data (from the VBN camera) collected in flight and IPM simulation/data analysis performed on the ground using Matlab. The algorithm processed the video frame by frame and extracted horizon and the runway from each frame. The roll and pitch of the aircraft were computed based on the horizon detected in each frame. The algorithm also identified the location of runway centreline in each frame which was further used to calculate the deviation of the aircraft from the runway centreline. Kalman filter was employed to reduce the effect of errors in the measurements. The roll and roll-rate results obtained for 800 frames are shown in Fig. 9. Similarly, Fig. 10 depicts the results for pitch and pitch-rate. The computed location of centreline (pixels) and the centreline drift rate (pixels per second) are shown in Fig.11.

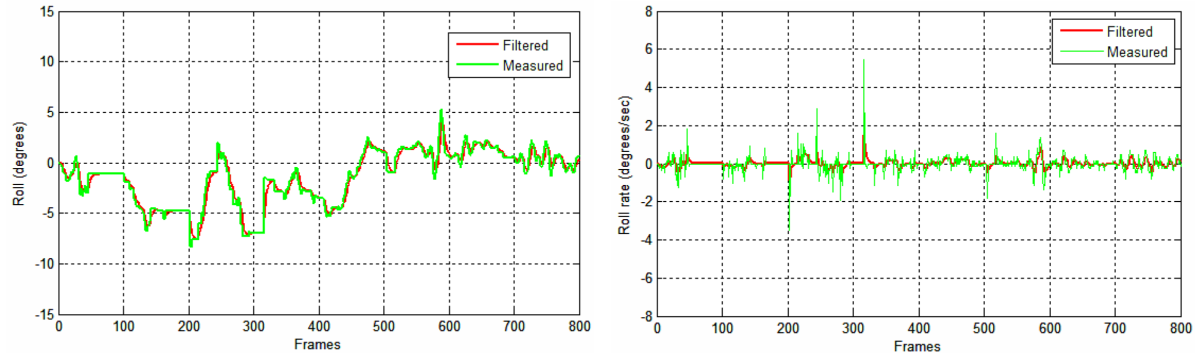

Figure 9: Roll and roll-rate computed from the test video.

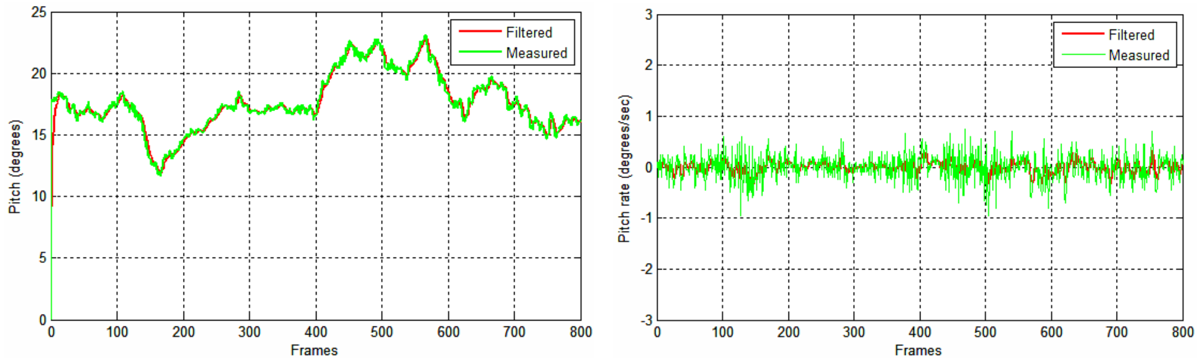

Figure 10: Roll and roll-rate computed from the test video.

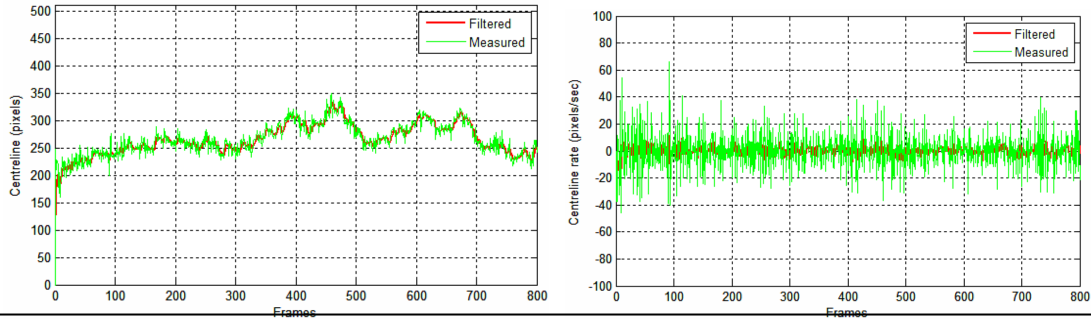

Figure 11: Computed location and rate of change of centreline location from the test video.

Although the test activities were carried out in a limited portion of the aircraft/UAV operational flight envelopes, some preliminary error analysis was performed comparing the performance of the VBN sensor an INS. The mean and standard deviation of the VBN attitude and attitude-rate measurements are listed in Table 2.

Table 2: VBS Attitude and Angular Rates Errors Parameters.

VBN Measured Parameters	Mean	Standard Deviation
Roll angle	0.22°	0.02°
Pitch angle	-0.32°	0.06°
Yaw angle (centreline deviation)	0.64°	0.02°
Roll Rate	0.33°/s	0.78 °/s
Pitch Rate	-0.43°/s	0.75°/s
Yaw Rate	1.86°/s	2.53°/s

The performance of the VBN sensor is strongly dependent on the characteristics of the employed camera. The developed algorithms are unable to determine the attitude of the aircraft in case of absence of horizon in the image. Similarly, the deviation of the aircraft from the runway centreline cannot be computed in the absence of runway in the image. The most severe physical constrain is imposed by the FOV of the camera. The maximum vertical and horizontal FOVs of the Flea Camera are 57.6° and 77.6° respectively. Due to this limitation, the VBN sensor can compute a minimum pitch angle of -28.8° and a maximum of +28.8°. Additionally, environmental factors such as fog, night/low-light conditions or rain also affect the horizon/runway visibility and degrade the performance of the VBN system.

3. INTEGRATION CANDIDATE SENSORS

There are a number of limitations and challenges associated to the employment VBN sensors in UAV platforms. VBN is best exploited at low altitudes, where sufficient features can be extracted from the surrounding. The FOV of the camera is limited and, due to payload limitations, it is often impractical to install multiple cameras. When multiple cameras are installed, additional processing is required for data exploitation. In this case, also stereo vision techniques can be implemented. Wind and turbulence disturbances must be modelled and accounted for in the VBN processing. Additionally the performance of VBN can be very poor in low-visibility conditions (performance enhancement can be achieved employing infrared sensors as well). However, despite these limitations and challenges, VBN is a promising technology for small-to-medium size UAV navigation and guidance applications, especially when integrated with other low-cost and low-weight/volume sensors currently available. In our research, we developed an integrated NGS approach employing two state-of-the-art physical sensors: MEMS-based INS and GPS, as well as augmentation from Aircraft Dynamic Models (Virtual Sensor) in specific flight phases.

3.1 GNSS and MEMS-INS Sensors Characteristics

GNSS can provide high-accuracy position and velocity data using pseudorange, carrier phase, Doppler observables or various combinations of these measurements. Additionally, using multiple antennae suitably positioned in the aircraft, GNSS can also provide attitude data. In this research, we considered GPS Standards Positioning Service (SPS) pseudorange measurements for position and velocity computations. Additional research is currently being conducted on GPS/GNSS Carrier Phase Measurements (CFM) for attitude estimation. Tables 3 lists the position and velocity of state-of-the-art SPS GPS receivers. Position error parameters are from [18] and velocity error parameters are from [19], in which an improved time differencing carrier phase velocity estimation method was adopted. Typically, GPS position and velocity measurements are provided at a rate of 1 Hz.

Table 3: GPS position and velocity errors.

Errors	**Mean**	**Standard Deviation**
North Position Error (m)	-0.4	1.79
East Position Error (m)	0.5	1.82
Down Position Error (m)	0.17	3.11
North Velocity Error (mm/s)	0	3.8
East Velocity Error (mm/s)	0	2.9
Down Velocity Error (mm/s)	2.9	6.7

An Inertial Navigation System (INS) can determine position, velocity and attitude of a UAV based on the input provided by various kinds of Inertial Measurement Units (IMUs). These units include 3-axis gyroscopes, measuring the roll, pitch and yaw rates of the aircraft around the body-axis. They also comprise 3-axis accelerometers determining the specific forces in the inertial reference frame. In our research, we considered a strap-down INS employing low-cost MEMS Inertial Measurement Units (IMUs). MEMS-based IMUs are low-cost and low-weight/volume devices that represent an attractive alternative to high-cost traditional INS sensors, especially for general aviation or small UAVs applications. Additionally, MEMS sensors do not necessitate high power and the level of maintenance required is far lower than for high-end INS sensors [20]. The main drawback of these sensors is the relatively poor level of accuracy of the measurements that they provide. In our research, INS-MEMS errors are modeled as White Noise (WN) or as Gauss-Markov (GM) processes [21, 22]. Table 5 lists the MEMS-INS error parameters considered in our research.

Table 4. MEMS-INS error parameters.

IMU Error Parameters	**Error Models**
p gyro noise	WN (0.53 °/s)
q gyro noise	WN (0.45 °/s)
r gyro noise	WN (0.44 °/s)
x accelerometer noise	WN (0.013 m/s^2)
y accelerometer noise	WN (0.018 m/s^2)
z accelerometer noise	WN (0.010 m/s^2)
p gyro bias	GM (0.0552 °/s, 300 s)
q gyro bias	GM (0.0552 °/s, 300 s)
r gyro bias	GM (0.0552 °/s, 300 s)
x accelerometer bias	GM (0.0124 m/s^2, 300 s)
y accelerometer bias	GM (0.0124 m/s^2, 300 s)
z accelerometer bias	GM (0.0124 m/s^2, 300 s)
p gyro scale factor	GM (10000 PPM, 18000 s)
q gyro scale factor	GM (10000 PPM, 18000 s)
r gyro scale factor	GM (10000 PPM, 18000 s)
x accelerometer scale factor	GM (10000 PPM, 18000 s)
y accelerometer scale factor	GM (10000 PPM, 18000 s)
z accelerometer scale factor	GM (10000 PPM, 18000 s)

3.3 ADM Virtual Sensor Characteristics

The ADM Virtual Sensor is essentially a knowledge-based module used to augment the navigation state vector by predicting the UAV flight dynamics (aircraft trajectory and attitude motion). The ADM can employ either a 6-Degree of Freedom (6-DOF) or a 3-DOF variable mass model with suitable constraints applied in the different phases of the UAV flight. The input data required to run these models are made available from aircraft physical sensors (i.e., aircraft data network stream) and form ad-hoc databases. Additionally, for the 3-DOF case, an automatic manoeuvre recognition module is implemented to model the transitions between the different UAV flight phases. Typical ADM error parameters are listed in Table 6 [21, 22]. Table 6 lists the associated error statistics obtained in a wide range of dynamics conditions for 20 seconds runtime.

Table 5: ADM error parameters.

ADM Error Parameters	Error Models
Coefficients (on all except the flap coefficients)	GM(10%,120s)
Control input	WN(0.02°) aileron, rudder, elevator
Center of gravity error [x,y,z]	Constant [0.001, 0.001, 0.001]m
Mass error	2% of true
Moment of inertia error [Jx,Jy,Jz,Jxz]	2% of true
Thrust Error	Force, 5% of true, Moment 5% of true
Gravity Error 1 σ	36 μg
Air Density Error	5% of true
Speed of sound Error	5% of true

Table 6: ADM error statistics.

	Mean	Standard Deviation
North Velocity Error	4.48E-3	3.08E-2
East Velocity Error	-3.73E-2	1.58E-1
Down Velocity Error	-4.62E-2	5.03E-2
Roll Error	4.68E-5	7.33E-3
Pitch Error	3.87E-3	2.41E-3
Yaw Error	-1.59E-3	7.04E-3

4. MULTISENSOR SYSTEM DESIGN AND SIMULATION

The data provided by all sensors are blended using suitable data fusion algorithms. Due to the non-linearity of the sensor models, an EKF was developed to fuse the information provided by the different MINS sensors and to provide estimates of position, velocity and attitude of the platform in real-time. Two different integrated navigation system architectures were defined, including VBN/IMU/GPS (VIG) and VIG/ADM (VIGA). The VIG architecture uses VBN at 20 Hz and GPS at 1 Hz to augment the MEMS-IMU running at 100 Hz.The VIGA architecture includes the ADM (computations performed at 100 Hz) to provide attitude channel augmentation. The corresponding VIG and VIGA integrated navigation modes were simulated using MATLAB^{TM} covering all relevant flight phases of the AEROSONDE UAV (straight climb, straight-and-level flight, straight turning, turning descend/climb, straight descent, etc.). The navigation system outputs were fed to an hybrid Fuzzy-logic/PID controller designed at Cranfield University for the AEROSONDE UAV and capable of operating with stand-alone VBN, as well as with VIG/VIGA and other sensors data [23, 24].

4.1VIG and VIGA Architectures

The VIG architecture is illustrated in Fig. 12. The INS position and velocity provided by the navigation processor are compared to the GPS position and velocity to form the measurement input of the data fusion block containing the EKF. A similar process is also applied to the INS and VBN attitude angles, whose differences are incorporated in the EKF measurement vector. The EKF provides estimates of the Position, Velocity and Attitude (PVA) errors, which are then removed from the sensor measurements to obtain the corrected PVA states. The corrected PVA and estimates of accelerometer and gyroscope biases are also used to update the INS raw measurements.

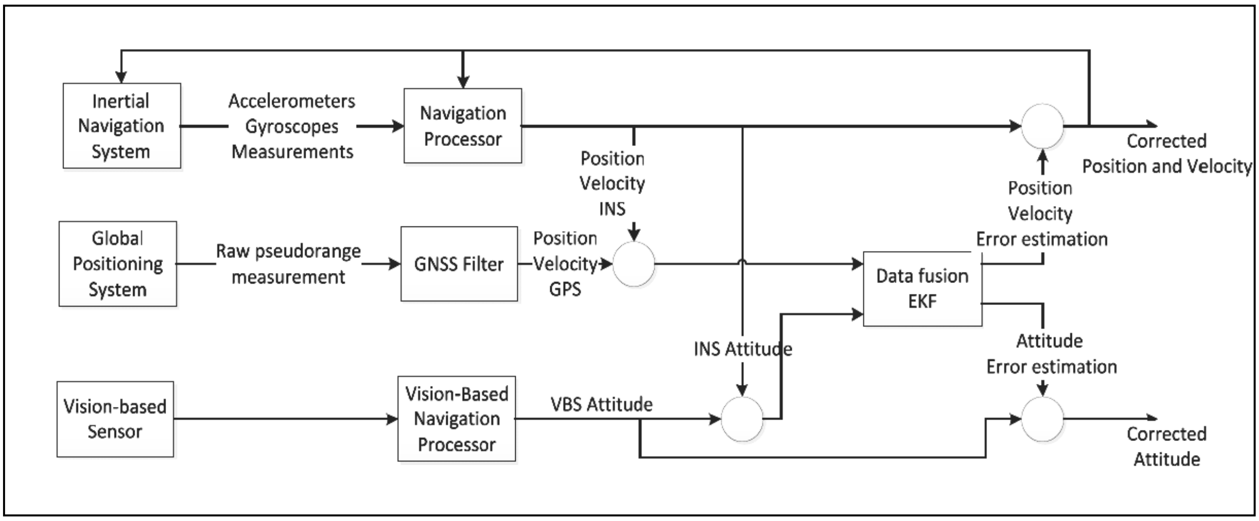
Inertial Navigation System
Accelerometers Gyroscopes Measurements
Navigation Processor
Position Velocity INS
Corrected Position and Velocity
Position Velocity Error estimation
Global Positioning System
Raw pseudorange measurement
GNSS Filter
Position Velocity GPS
Data fusion EKF
INS Attitude
Attitude Error estimation
Vision-based Sensor
Vision-Based Navigation Processor
VBS Attitude
Corrected Attitude

Figure 12. VIG Architecture.

The VIGA architecture is illustrated in Fig. 13. As before, the INS position and velocity provided by the navigation processor are compared to the GPS data to form the measurement input of EKF. Additionally, in this case, the attitude data provided by the ADM and the INS are compared to feed the EKF at 100 Hz, and the attitude data provided by the VBS and INS are compared at 20 Hz and input to the EKF. Like before, the EKF provides estimations of PVA errors, which are removed from the INS measurements to obtain the corrected PVA states. Again, the corrected PVA and estimates of accelerometer and gyroscope biases are used to update INS raw measurements.

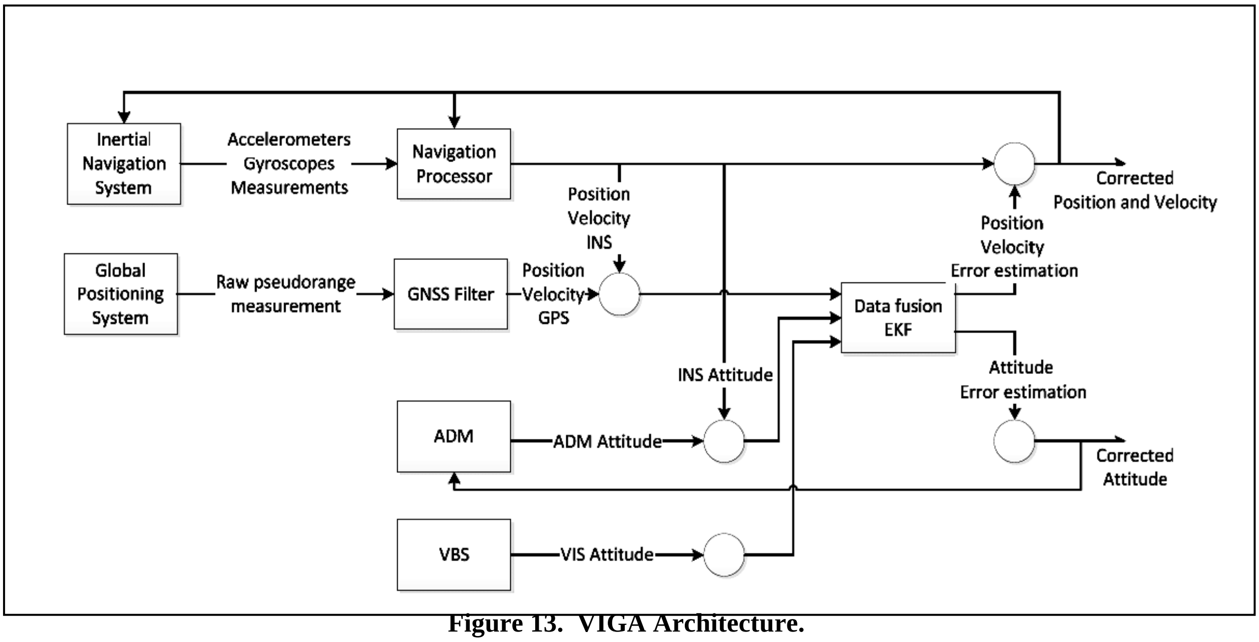
Inertial Navigation System
Accelerometers Gyroscopes Measurements
Navigation Processor
Position Velocity INS
Corrected Position and Velocity
Position Velocity Error estimation
Global Positioning System
Raw pseudorange measurement
GNSS Filter
Position Velocity GPS
Data fusion EKF
INS Attitude
Attitude Error estimation
ADM
ADM Attitude
Corrected Attitude
VBS
VIS Attitude

Figure 13. VIGA Architecture.

5. VIG and VIGA Simulation

Both the VIG and VIGA multisensor architectures were tested by simulation in an appropriate sequence of flight manoeuvres representative of the AEROSONDE UAV operational flight envelope. An FLC/PID controller was used for simulation. The duration of the simulation is 1150 seconds (approximately 19 minutes). The horizontal and vertical flight profiles are shown in Fig. 14.

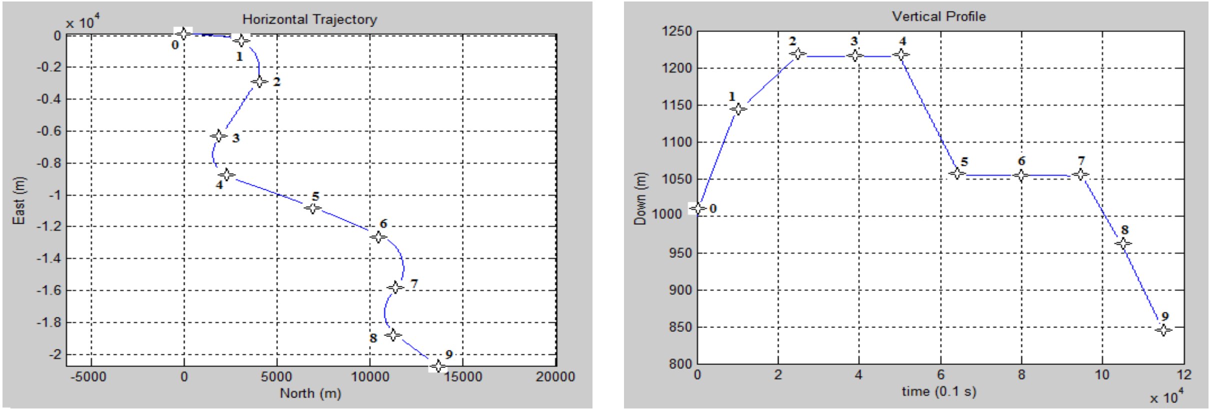

Figure 14: Horizontal and vertical flight profiles.

The list of the different simulated flight manoeuvres and associated control inputs is provided in Table 9. The numbered waypoints are the same shown in Fig. 14.

Table 7. Flight manoeuvres and control inputs.

Flight Maneuver	Required Roll (°)	Required Pitch (°)	Time (s)	Legs (Waypoints)
Straight Climb (Take off)	0	10.00	100	[0,1]
Right Turning Climb	-2	4.00	150	[1,2]
Straight and Level	0	2.25	150	[2,3]
Level Left Turn	3	2.25	100	[3,4]
Straight Descent	0	-0.7	150	[4,5]
Straight and Level	0	2	150	[5,6]
Level Right Turn	-2	2	150	[6,7]
Left Turning Descent	3	-0.5	100	[7,8]
Straight Descent	0	-1	100	[8,9]

The VIG position error time histories (east, north and down) are shown in Fig. 15. For comparison, also the GPS position errors (unfiltered) are shown. Table 8 presents the position error statistics associated to each flight phase.

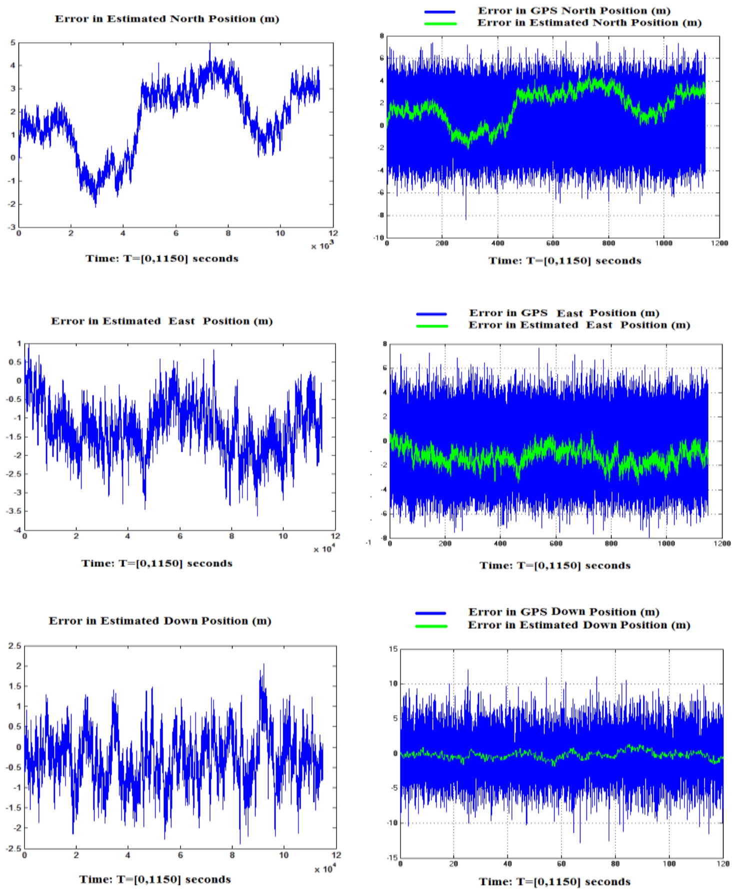

Figure 15. VIG position error time histories.

Table 8 : VIG position error stastics.

Phase of Flight	North Position		East Position		Down Position	
	Mean (m)	σ (m)	Mean (m)	σ (m)	Mean (m)	σ (m)
Straight Climb	1.22	4.38E-01	-4.34E-01	6.20E-01	-1.80E-01	5.30E-01
Right Turning Climb	9.88E-01	8.68E-01	-1.36	4.86E-01	-2.40E-01	6.02E-01
Straight and Level	-8.01E-01	4.71E-01	-1.51	5.02E-01	-4.38E-01	6.82E-01
Level Left Turn	1.16	1.32	-1.66	5.68E-01	-5.28E-01	6.57E-01
Straight Descent	2.77	3.60E-01	-7.20E-01	4.48E-01	-4.92E-01	6.36E-01
Level Right Turn	2.06	1.07	-1.86	5.92E-01	-1.29E-01	8.37E-01
Left Turning Descent	1.65	7.29E-01	-1.63	5.35E-01	-2.66E-01	6.23E-01

The VIG velocity error time histories are shown in Fig. 16. For comparison, the GPS velocity error time histories are also shown. GPS is the dominating sensor for velocity computations but a significant improvement is with the VIG system on the accuracy of the vertical data. Table 9 shows the velocity error statistics associated to each flight phase.

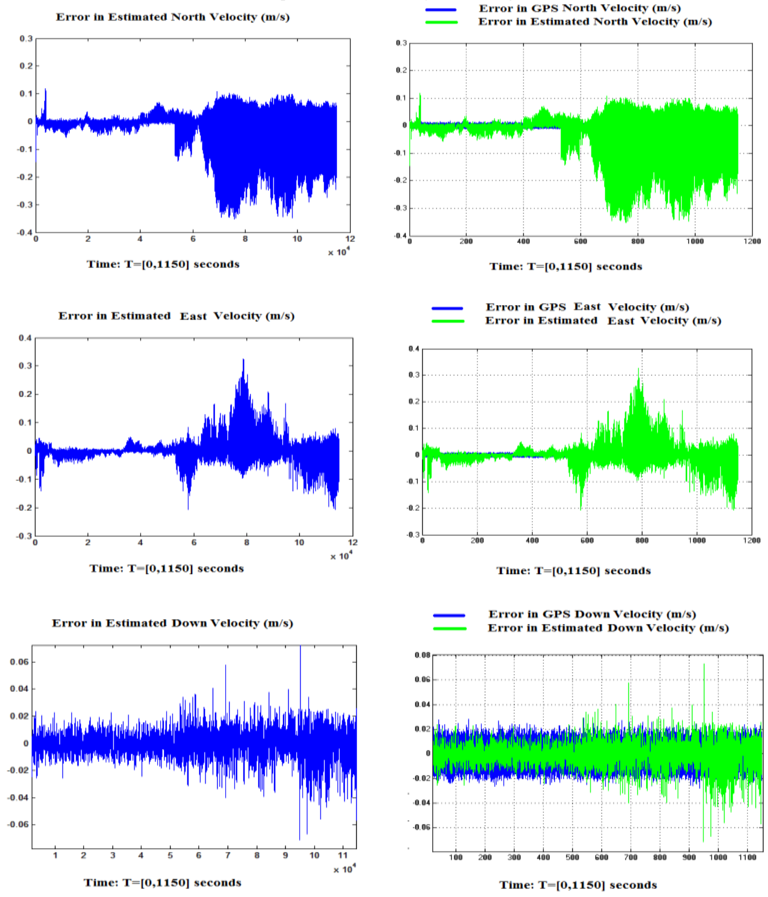

Figure 16. VIG velocity error time histories.

Table 9: VIG velocity error statistics.

Phase of Flight	North Velocity		East Velocity		Down Velocity	
	Mean (m/s)	σ (m/s)	Mean (m/s)	σ (m/s)	Mean (m/s)	σ (m/s)
Straight Climb	-6.40E-03	1.73E-02	-4.14E-03	2.14E-02	1.30E-02	1.89E-01
Right Turning Climb	-7.97E-03	1.11E-02	-7.59E-03	7.93E-03	-2.90E-04	6.79E-03
Straight and Level	-7.19E-03	1.00E-02	3.63E-03	1.08E-02	-4.20E-04	6.78E-03
Level Left Turn	1.44E-02	1.51E-02	3.61E-03	6.99E-03	-1.80E-04	6.90E-03
Straight Descent	-4.50E-03	3.64E-02	-3.71E-03	3.32E-02	-3.80E-04	9.55E-03
Level Right Turn	-3.11E-02	1.08E-01	1.21E-02	4.91E-02	7.64E-04	9.82E-03
Left Turning Descent	-3.28E-02	1.13E-01	-6.33E-03	3.71E-02	-1.32E-03	1.54E-02

The attitude error time histories of the VIG system are shown in Fig. 17. Table 10 presents the associated attitude error statistics.

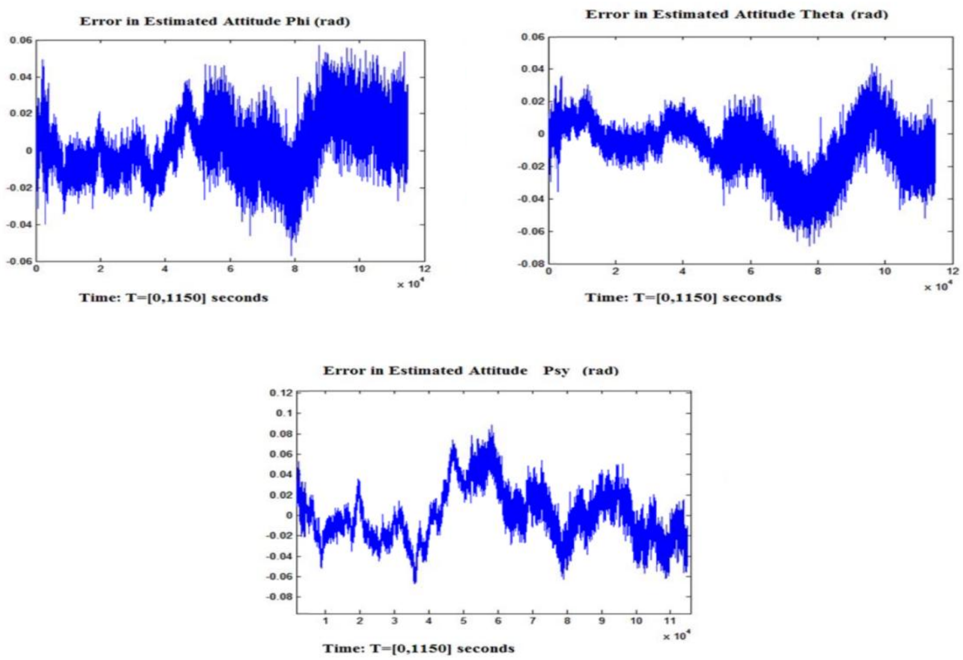

Figure 17. Attitude error time histories.

Table 10: VIG attitude error statistics.

Phase of Flight	**Roll (Phi)**		**Pitch (Theta)**		**Yaw (Psy)**	
	Mean (m/s)	**σ (m/s)**	**Mean (m/s)**	**σ (m/s)**	**Mean (m/s)**	**σ (m/s)**
Straight Climb	-6.07E-02	5.54E-01	3.16E-01	2.92E-01	-2.17E-01	1.06
Right Turning Climb	-4.25E-01	3.09E-01	-5.10E-02	4.65E-01	-5.39E-01	6.99E-01
Straight and Level	-4.22E-01	3.44E-01	-2.24E-02	3.80E-01	-1.38	8.02E-01
Level Left Turn	6.13E-01	4.96E-01	-1.39E-01	4.27E-01	1.39	1.36
Straight Descent	3.89E-01	4.60E-01	-3.68E-01	3.50E-01	2.03	1.08
Level Right Turn	7.58E-01	7.69E-01	-5.46E-01	8.54E-01	3.91E-01	8.59E-01
Left Turning Descent	1.22	7.06E-01	-5.37E-01	7.84E-01	-3.36E-01	8.86E-01

As discussed above, the ADM data were used in the VIGA architecture to update the attitude channel (the position and velocity channels are derived from the VIG system). Therefore, only the attitude error statistics of the VIGA system are presented here. The time histories of the VIGA attitude errors are shown in Fig. 18 and compared with the corresponding VIG attitude errors. Table 11 presents the VIGA attitude error statistics.

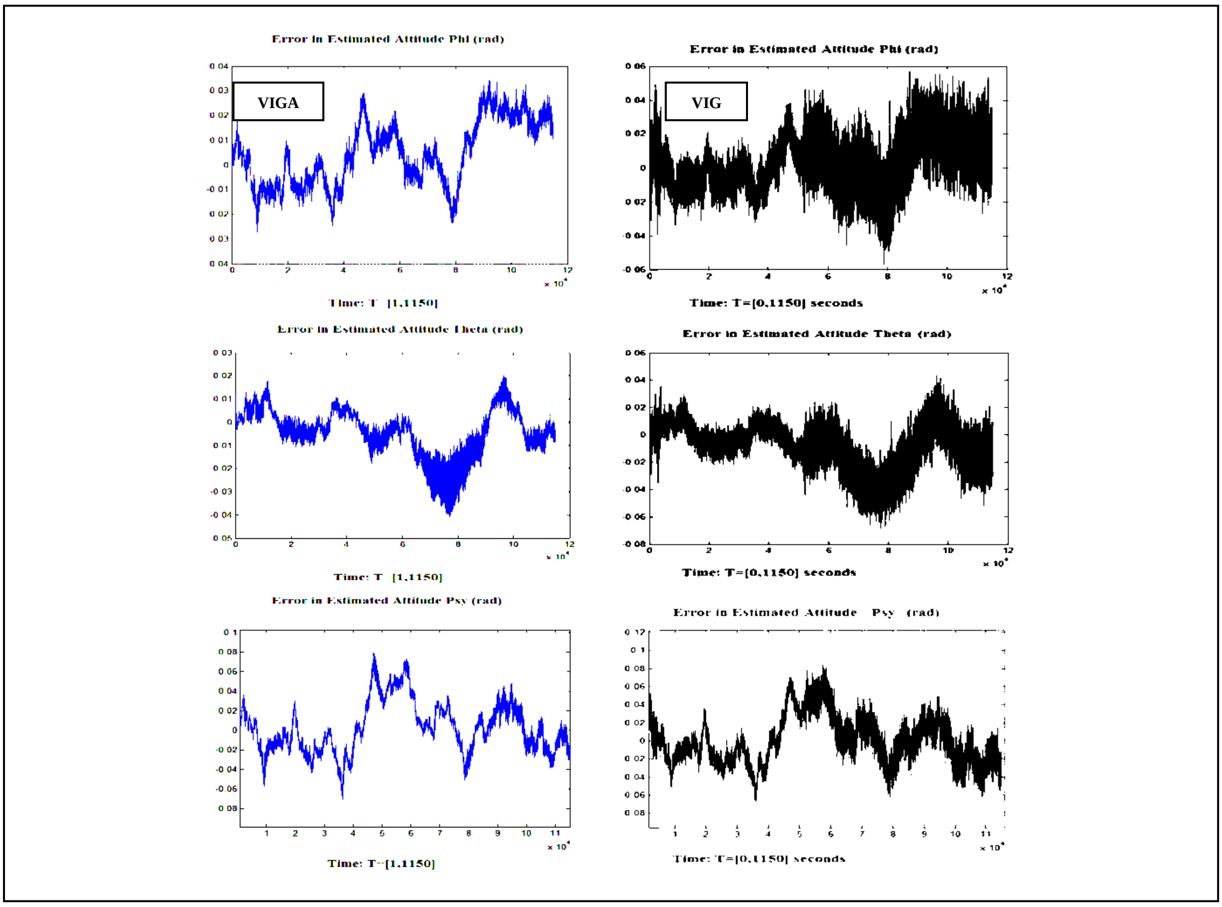

Figure 18. Comparison of VIGA and VIG attitude errors.

Table 11: VIGA attitude error statistics.

Phase of Flight	Roll (Phi)		Pitch (Theta)		Yaw (Psy)	
	Mean (deg)	σ (deg)	Mean (deg)	σ (deg)	Mean (deg)	σ (deg)
Straight Climb	-6.76E-02	5.19E-01	3.58E-01	2.08E-01	-1.19E-01	1.01
Right Turning Climb	-4.42E-01	2.64E-01	-8.37E-02	4.07E-01	-6.07E-01	6.97E-01
Straight and Level	-4.38E-01	3.06E-01	-3.61E-02	3.27E-01	-1.44	7.92E-01
Level Left Turn	6.16E-01	4.77E-01	-1.60E-01	3.70E-01	1.45	1.31
Straight Descent	3.92E-01	3.58E-01	-4.22E-01	2.37E-01	2.07	1.08
Level Right Turn	7.79E-01	6.99E-01	-6.49E-01	7.39E-01	4.73E-01	8.30E-01
Left Turning Descent	9.00E-02	1.44E-01	3.74E-01	4.50E-01	-1.78E-01	8.59E-01

During the initial VIGA simulation runs it was evidenced that the ADM data cannot be used without being reinitialised regularly. For the AEROSONDE UAV manoeuvres listed in Table 10, it was found that the optimal period between ADM reinitialisation was in the order of 20 seconds. Converting the data in Table 10 and 11 to the corresponding RMS (95%) values, we obtain the error data in Table 12 and 13. Comparing the two tables, it is evident that the ADM virtual sensor contributes to a moderate reduction of the overall attitude error budget in all relevant flight phases.

Table 12: VIG attitude RMS-95% errors.

Phase of Flight	RMS-95% Phi (deg)	RMS-95% Theta (deg)	RMS-95% Psy (deg)
Straight Climb	1.11	9.24E-01	2.13
Right Turning Climb	1.05	9.44E-01	1.85
Straight and Level	1.09	7.63E-01	3.30
Level Left Turn	1.58	9.13E-01	3.97
Straight Descent	1.20	1.10	4.67
Level Right Turn	2.16	2.14	1.96
Left Turning Descent	2.46	1.90	1.89

Table 13: VIGA attitude RMS-95% errors.

Phase of Flight	RMS-95% Phi (deg)	RMS-95% Theta (deg)	RMS-95% Psy (deg)
Straight Climb	1.05	7.56E-01	1.97
Right Turning Climb	1.03	8.20E-01	1.76
Straight and Level	1.07	6.55E-01	3.18
Level Left Turn	1.56	7.90E-01	3.82
Straight Descent	1.06	8.76E-01	4.60
Level Right Turn	2.09	1.84	1.84
Left Turning Descent	1.42	1.17	1.76

To conclude the simulation data analysis, Table 14 shows a comparison of the VIG/VIGA horizontal and vertical accuracy (RMS-95%) with the required accuracy levels for precision approach [25, 26]. The VIG/VIGA performances are in line with CAT II precision approach . requirements. Future research will address the possible synergies of the VIG/VIGA architectures with GPS/GNSS space, ground and aircraft-based augmentation systems

Table 14: VIGA attitude RMS-95% errors.

Category of approach	Horizontal Accuracy (m) 2D RMS-95%		Vertical Accuracy (m) RMS-95% Down	
	Required	VIG/VIGA	Required	VIG/VIGA
CAT I	16	5,8	4	1.61
CAT II	6.9		2	
CAT III	4.1		2	

6. CONCLUSIONS

In this paper we have described the research activities performed to design a low-cost and low-weight/volume integrated NGS system suitable for small/medium size UAV applications. As a first step, we designed and tested a VBN sensor employing appearance-based techniques and specifically tailored for UAV low-level flight, including precision approach and landing operations. In addition to horizon detection and image-flow, the VBN sensor also employed runway features extraction during the approach phase. Various candidates were considered for integration with the VBN sensor and, as a result, GPS and MEMS-IMUs (with possible augmentation from ADM) were finally selected. The multisensory integration was accomplished with an EKF. The attitude/attitude-rate accuracies obtained with the VBN sensor were evaluated by a combination of laboratory, ground and flight test activities. The results were satisfactory in low-level flight and during the approach and landing phase of the UAV flight. However, the VBN sensor performance was strongly dependent on the characteristics of the employed camera. The algorithms developed are unable to determine the attitude of the aircraft in case of absence of horizon in the image. Similarly, the deviation of the aircraft from the runway centreline cannot be computed in the absence of runway in the image. The most severe physical constrain is imposed by the angular FOV of the camera. The maximum vertical and horizontal FOVs of the employed camera are 57.6° and 77.6° respectively. Due to this limitation, the VBN sensor can compute a minimum pitch angle of -28.8° and a maximum of +28.8°. Current research shows that for wind speeds greater than 20m/s, the VBN/IPM algorithms are marginally usable for navigation purposes as well as for guidance. Environmental factors such as fog, night/low-light conditions or rain also affect the horizon/runway visibility and degrade the performance of the VBN system. To cope with these limitations, current research is investigating the potential synergies obtained by integrating daylight camera vision sensors with Infrared and Night Vision Imaging Sensors (IR/NVIS). Simulation of the VIG integrated navigation mode showed that this integration scheme can achieve horizontal/vertical position accuracies in line with CAT-II precision approach requirements, with a significant improvement compared to stand-alone SPS GPS. An improvement was also observed in the accuracy of the vertical velocity data. Additionally, simulation of the VIGA navigation mode showed promising results since, in most cases, the attitude accuracy is higher using the ADM/VBS/IMU rather than using VBS/IMU only. However, due to rapid divergence of the ADM virtual sensor, there is a need for a frequent re-initialisation of the ADM data module, which is strongly dependent on the UAV flight dynamics and the specific manoeuvres/flight-phase transitions performed. In the considered portion of the UAV operational flight envelope, the required re-initialisation interval was approximately 20 seconds. To cope with this issue, the original ADM is being modified to take into account specific manoeuvre constraints and the transition states between various manoeuvres are being carefully modelled. Additionally, an automatic manoeuvre recognition algorithm is being developed for updating the ADM in real-time and providing direct feedback to the VBN sensor. It is expected that, adopting this approach, the performances of the multisensory integrated NGS will be significantly enhanced both in terms of data accuracy and data continuity.

REFERENCES

[1] Sinopoli, B., Micheli, M., Donato, G. and Koo, T.J., “Vision based navigation for unmanned aerial vehicles,” Proceeding of International Conference of Robotics & Automation, Vol. 2, 2001, pp. 1757 – 1764.

[2] Se S., Lowe D.G. and Little J.J., “Vision based global localisation and mapping,” IEEE Trans. on Robotics, Vol. 21, No.3, June 2005, pp. 364-375.

[3] Cui, P. and Yue, F., “Stereo vision-based autonomous navigation for lunar rovers,” Aircraft Engineering and Aerospace Technology: An International Journal, Vol. 79, No. 4, 2007, pp. 398-405.

[4] Matsumoto Y., Sakai K., Inaba, M. and Inoue, H., “View-based approach to robot navigation,” proceedings of the 2000 IEEE/RSJ conference on Intelligent Robots and Systems, Vol. 3, Nov. 2000, pp. 1702-1708.

[5] Courbon, J., Mezouar, Y., Guenard, N. and Martinet, P., “Visual navigation of a quadrotor aerial vehicle,” proceedings of the 2009 IEEE/RSJ conference on Intelligent Robots and Systems, Oct. 2009, pp. 5315-5320.

[6] Courbon, J., Mezouar, Y., Guenard, N. and Martinet, P., ”Vision-based navigation of unmanned aerial vehicles,” Control Engineering Practice, Vol. 18, Issue 7, July 2010, pp. 789-799.

[7] Chen, Z. and Birchfield, S.T., “Qualitative vision-based path following,” IEEE Trans. on Robotics, Vol. 25, Issue 3, June 2009, pp. 749-754.

[8] Remazeilles, A., Chaumette, F., “Image-based robot navigation from an image memory,” Journal of Robotics and Autonomous Systems, Vol. 55, Issue 4, 2007.

[9] Xinhua, L. and Cao, Y., “Research on the application of the vision-based autonomous navigation to the landing of the UAV,” proceedings of Fifth International Symposium on Instrumentation and Control Technology, Vol. 5253, 2003, pp. 385-388.

[10] Dusha, D., Mejias, L. and Walker, R., “Fixed-wing attitude estimation using temporal tracking of the horizon and optical flow,” Journal of Field Robotics, Vol. 28, No. 3, 2011, pp. 355-372.

[11] Olivares-Mendez, M.A., Mondragon, I.F., Campoy, P. and Martinez, C., “Fuzzy controller for UAV-landing task using 3D position visual estimation,” proceedings of IEEE International Conference on Fuzzy Systems, 2010.

[12] Roumeliotis, S.I., Johnson, A.E. and Montgomery, J.F., “Augmenting Inertial Navigation with image-based estimation,” proceeding of International Conference of Robotics & Automation, 2002, pp. 4326-4333.

[13] Desouza, G.N., Kak, A.C., “Vision for mobile robot navigation: a survey,” IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 24, Issue 2, Feb. 2002, pp. 237 – 267.

[14] Santosh, D., Achar, S. and Jawahar, C.V., “Autonomous image-based exploration for mobile robot navigation,” proceeding of International Conference of Robotics & Automation, 2008, pp. 2717 – 2722.

[15] Rives, P. and Azinheira, J.R., “Visual auto-landing of an autonomous aircraft,” INRIA, No. 4606, 2002.

[16] Blanc, G., Mezouar, Y. and Martinet, P., “Indoor navigation of a wheeled mobile robot along visual routes,” proceeding of International Conference of Robotics & Automation, 2005, pp. 3354-3359.

[17] Rangasamy, G., “Image sensor fusion algorithms for obstacle detection, location and avoidance for autonomous navigation of UAVs,” M.Sc. Thesis, School of Engineering, Cranfield University, 2010.

[18] Tiberius, C. (2003). Standard positioning service, Handheld GPS receiver acuracy. GPS World.

[19] Ding, W., & Wang, J., "Precise Velocity Estimation with a Stand-Alone GPS receiver". University of New South Wales. Journal of the Institute of Navigation (USA), 2011.

[20] Titterton, D., & Weston, J., "Strapdown Inertial Navigation Technology" (2nd Edition). The Institution of Electrical Engineers, 2004.

[21] Troy, S., Bruggemann, Eng (HONS), B., & Eng, M., "Investigation of MEMS Inertial Sensors and Aircraft Dynamic Models in Global Positioning System Integrity Monitoring for Approaches with Vertical Guidance." Unpublished PhD thesis, Queensland University of Technology, School of Engineering, 2009.

[22] Godha, S., "Performance Evaluation of Low Cost MEMS-Based IMU Integrated With GPS for Land Vehicle Navigation Application." UCGE Report No. 20239, University of Calgary, Department of Geomatics Engineering, Alberta, Canada, 2006.

[23] Sabatini, R., Kaharkar, A., Shaid, T., Bartel, C., Jia, H., Zammit-Mangion, D., "Vision-based Sensors and Integrated Systems for Unmanned Aerial Vehicles Navigation and Guidance." Proceedings of the SPIE Conference Photonics Europe 2012. Brussels (Beligium), April 16-19, 2012.

[24] Sabatini, R., Leopoldo Rodríguez, Kaharkar, A., Bartel, C., Shaid, T., "GNSS Data Processing for Attitude Determination and Control of Unmanned Aerial and Space Vehicles." Proceedings of the European Navigation Conference 2012. Gdansk (Poland), April 25-27, 2012.

[25] ICAO - Annex 10 to the Convention on International Civil Aviation. “Aeronautical Telecommunications - Volume 1: Radio Navigation Aids.” Edition 6. July 2006.

[26] CAA Safety Regulation Group Paper 2003/09. “GPS Integrity and Potential Impact on Aviation Safety”. 2003.