DOI 10.1007/s10846-014-0043-8

Efficient Terrain-Aided Visual Horizon Based Attitude Estimation and Localization

Steven J. Dumble · Peter W. Gibbens

Received: 7 April 2013 / Accepted: 24 February 2014 / Published online: 19 March 2014

Abstract Inertial Navigation Systems typically rely on aiding-sensors such as GPS (Global Positioning System) to estimate the location of the system. The navigational performance of the Inertial Navigation System can be severely degraded when the GPS measurements are inaccurate or unavailable. Terrain-Aided Navigation is another method of localizing the platform by correlating the measured terrain information with a Digital Terrain Model. This paper presents an efficient Terrain-Aided Navigation method of generating position measurements from the visual appearance of the horizon profile (and hence terrain) surrounding the platform. An optimization process is used to align the measured horizon profile to an off-line pre-generated terrain-aided reference profile which allows for efficient position and attitude estimation. Numerical simulations are presented to evaluate the effectiveness of the proposed method. These results show that precise real-time attitude and position estimation is achievable using visual horizon profile information.

Keywords Terrain reference navigation · Unmanned aerial vehicle · Vision based localization · Digital terrain model

S. J. Dumble (✉) · P. W. Gibbens
University of Sydney, Sydney, New South Wales, Australia
e-mail: steven.dumble@sydney.edu.au

P. W. Gibbens
e-mail: peter.gibbens@sydney.edu.au

1 Introduction

Modern aircraft and Unmanned Aerial Vehicle (UAV) navigation systems usually consist of a sensor-aided Inertial Navigation System (INS) [1–3]. These systems integrate high-rate inertial measurements from an Inertial Measurement Unit (IMU) with respect to time to provide continuous position and attitude estimates of the platform. These integrated inertial estimates are accurate over small time intervals but they are subject to inertial drift over the longer term. This dead-reckoning or inertial error needs to be corrected or constrained by fusing measurements from an aiding sensor, hence sensor-aided INS. This fusion process allows accurate navigation over the long term operation of the system if the aiding measurements are continuously fused.

GNSS (Global Navigation Satellite System) or GPS (Global Positioning System) measurements are one of the main aiding sensors used in commercial INS units [1, 4–6]. However GNSS and GPS units require external infrastructure and radio signals to generate the aiding measurements. These signals can be jammed or spoofed in military applications when exposed to Electronic Countermeasure (ECM) conditions or they can be subject to multipath errors or obstructions which are common in maneuvering flight. The measurement precision can also be reduced by the current visible satellite constellation geometry. In such GPS-denied or limited situations, the navigational precision of the navigation system can drop dramatically as it then depends upon the accuracy of the inertial sensors which are subject to inertial drift.

Terrain-Aided Navigation (TAN) or Terrain Referenced Navigation (TRN) [1, 3] is a method of generating position aiding measurements which do not rely on external signals being transmitted to the aircraft, though it may depend on using active sensors. These systems can be used to increase the autonomy and reliability of the navigation systems. TAN systems use the concept of correlating measured terrain information with a terrain spatial database to provide position measurements. This is particulary important to autonomous or covert aerial operations.

This paper presents an efficient TAN method of generating position measurements from the visual appearance of the horizon profile. The use of visual sensors (or other spectrums) means that the TAN system is completely passive. The vision system presented in this paper assumes that the complete horizon surrounding the platform is visible by combining multiple cameras together to form a complete 360° Field of View (FOV). The use of the complete horizon profile is a feasible assumption as multiple camera and omnidirectional camera vision systems have been shown to be effective for horizon based attitude determination [7]. Correlating observed terrain features with a terrain database can be computationally expensive [8–15]. However, the method presented in this paper achieves computational efficiency by using an efficient method to generate the estimated horizon profile for any position within the domain of the terrain database. This is achieved by using off-line pre-computed reference horizon profiles which can be transformed and matched to the observed horizon profile using an optimization process. Since the reference profiles can be pre-computed, this dramatically decreases the computational burden as only a few hundred profile reference points need to be transformed to obtain the current profile; rather than rendering the whole terrain map which could consist of millions of terrain points. Thus, real-time performance can be realized without the need of specialized computational hardware such as Graphical Processing Units (GPUs).

This paper is organized as follows; Section 2 reviews Terrain Referenced Navigation. Section 3 describes the proposed visual horizon TAN method. Section 4 presents simulation results and discussion. Finally, the conclusion and future work are described in Section 5.

2 Terrain-Referenced Navigation

A mainstream implementation of TAN is Terrain Contour Matching (TERCOM) which was initially applied for cruise missile guidance [16]. The TERCOM system correlates radar clearance measurements of the terrain with a terrain map for position estimation. As this system relies on radar measurements, it is an active system which can be detected and jammed. A similar TAN method is TERPROM (Terrain Profile Matching). This system has been widely accepted in the aerospace industry. The active sensors that are typically used in TERPROM are infrared, optical, laser and millimeter wave radar [17]. The use of active sensors means that TERCOM and TERPROM cannot be fully utilized for military applications in hostile environments without the risk of detection. This risk is usually minimized by directing the energy downwards towards the ground to limit the revealing effect.

The use of visual measurements to recover the terrain height information is currently an active area of investigation [18–24]. Visual terrain elevation methods have advantages over radar-based methods of being cheap and completely passive such that they cannot be jammed or detected. These advantages are of great importance to military applications as active TAN sensors cannot be fully utilized in an hostile environment. Typically, the visual terrain measurements are made by feature correspondence in stereo image pairs or from optical flow. This allows the extrinsic properties of the terrain to be estimated from known camera displacements. Once the visual terrain elevation measurements have been made, the rest of the position estimation process follows a similar method to any other radar-based TAN system. Recent work has been undertaken [25–27] to use modern INS data-fusion techniques such as particle filtering to increase the accuracy of the position estimation process. The use of GPUs has also been investigated in such systems to make the system real-time realizable [28].

Terrain matching methods whether visual or radar elevation based have difficulties at lower altitudes as the terrain region visible under the aircraft is quite small. This makes it very difficult to correlate the observed terrain with any accuracy to a terrain map. However, at this low altitude condition, the horizon profile can provide strong positional information. To this end, the horizon profile has been investigated as another feature for visual navigation [8–15, 29, 30]. The use of the horizon profile as a navigation feature provides additional complications compared to direct terrain elevation correlation; especially to the computational burden imposed on the system. The horizon profile is heavily viewpoint dependent and contains no range information, such that the scale is lost. However, the horizon profile contains strong bearing information. Compounding these problems is the complex shape that the horizon profile can take and shear size of the combinatorics of the matching and localization problem [13].

A large amount of work in this area has been carried out for planetary/lunar rovers [10], where terrain data is available but where global positioning methods such as GPS are not. The work in this area has largely focused on the *drop-off* problem, where a rover is dropped off at an unknown location and it must localize itself without any *a priori* positional information. This is not as large a problem in the aeronautical industry as some estimate of the aircraft position is known at sometime during the flight. The main purpose of TAN systems for aeronautical applications is to keep the position or INS solution constrained rather than undertaking a complete positional search at each update step.

A feasibility study was presented in [29, 30], which described a checkpoint system for aircraft navigation in which the observed horizon profile is compared using the integral absolute differences to precomputed reference horizon profiles along the planned flight path.

More generalized horizon localization methods have been developed. Mountain peaks in the horizon profile are the most common feature used in these methods. The profile peaks are usually the most informative part of the horizon and the most viewpoint independent. The process used in [8–11] was to detect peaks in the horizon profile and then carry out an alignment process to align them to a pre-computed peak map. Mountain peaks and rigid lines from the entire image (not just the horizon outline) were also used in [14, 15] to help calculate the view location. Other horizon-based methods use a feature matching approach, which matches horizon curve segments to a pre-computed spatial view database. The horizon profile is split into characterizable curves in [12, 13] and then pre-computed tables can be searched to determine viewpoint positions with similar curves and distributions. The complete horizon profile was also used in different methods for fine alignment [8] and validation [9, 11] of the position estimate to increase accuracy and search efficiency.

The complex nature of the horizon localization problem means that the solution methods are computationally expensive; which makes them difficult to implement in a real-time system. The reported computation time for the different peak matching methods varies between 3 seconds [10] to 4 minutes [9, 11] when *a priori* position information is used. The horizon navigation method of precomputing and characterizing horizon curve segments [13] allows for a decreased computation time of approximately 1 minute without *a priori* position information. The reported position accuracies for these terrain-aided horizon navigation methods are approximately between 100 meters [10, 14, 15] and 400 meters [9, 11]. The more recent work on horizon profile matching undertaken in [31] reports a 3 minute computation time to estimate the attitude only while using a GPU to accelerate the computations. A recent combined position and attitude estimation approach [32] reported a computational time of 10 seconds and an accuracy of 1 km without the use of a GPU.

The method proposed in this paper treats the complete horizon profile as the feature to be matched rather than breaking the profile into peaks or curve segments. This helps to improve the positional accuracy as the whole profile is used to localize the platform rather than only using a subsection of the observed horizon.

3 Efficient Terrain-Aided Visual Horizon Based Localization

This proposed terrain-aided localization process takes visual measurements of the observed horizon in the image frame and aligns them to the terrain-aided horizon profile in the horizon frame by optimizing the estimated attitude and position of the platform. The generation of the terrain-aided horizon profile for any given position can be very computationally expensive and not suitable for a real-time optimization process. To increase the efficiency of this process, reference horizon profiles are generated off-line and an efficient transformation process is developed to shift the pre-generated horizon profile from the reference position to a nearby view location. Including this transformation process inside the optimization process allows for efficient terrain-aided horizon attitude and position localization. This transformation process also allows the analytical jacobians for each optimization iteration to be calculated, increasing the computational efficiency of the whole optimization process.

The measured attitude and location then can be fused with an INS to produce a terrain-aided INS solution. These measurements may also be fused with other sensors such as GPS using a data fusion algorithm to provide extra measurements and redundancies. The terrain aided localization measurements could keep the solution constrained while operating in GPS denied areas when suitable terrain and operating conditions are met. Even if the GPS is operating correctly, there is still added benefits of the horizon based system as it provides direct, highly accurate attitude measurements that can greatly reduce the attitude error. Single antenna GPS systems cannot measure the attitude directly and the accuracy of the attitude solution is of critical importance for guidance, navigation and control of aerospace platforms.

The background theory and development of the required processes for the optimization are described in the Subsections 3.1 – 3.6 and then an overview of the complete localization process is outlined in Subsection 3.7.

3.1 Horizon Profile Representation

Let $\mathbf{H}^h\big|_{\mathbf{P}_0}$ be the horizon profile expressed in the horizon frame $\mathcal{F}_h$ and evaluated for a specific viewpoint position $\mathbf{P}_0$. The horizon frame $\mathcal{F}_h$ is a polar representation of the NED (North, East, Down) navigational frame $\mathcal{F}_n$, such that a horizon point $\mathbf{P}^n$ in the navigational frame $\mathcal{F}_n$ (relative to the origin at $\mathbf{P}_0$) can be expressed in the horizon frame $\mathcal{F}_h$ using:

$$\mathbf{P}^h = \begin{bmatrix} \psi_h \\ \theta_h \\ R_h \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{\mathbf{P}^n_y}{\mathbf{P}^n_x}\right) \\ -\arctan\left(\frac{\mathbf{P}^n_z}{R_h}\right) \\ \sqrt{(\mathbf{P}^n_x)^2 + (\mathbf{P}^n_y)^2} \end{bmatrix} \tag{1}$$

where ψ_h, θ_h and R_h are the horizon bearing, horizon pitch and horizon range respectively and $\mathbf{P}^n_x$, $\mathbf{P}^n_y$ and $\mathbf{P}^n_z$ are the individual axis components of $\mathbf{P}^n$.

Let the horizon profile $\mathbf{H}^h\big|_{\mathbf{P}_0}$ be parameterized into m ordered, equally spaced bearing data points ψ_{h_i} such that $\psi_{h_1} < \psi_{h_2} < \cdots < \psi_{h_m}$ with $(\psi_{h_{i+1}} - \psi_{h_i}) = \Delta\psi_h$. The data set spans the complete 360° horizon profile so $\psi_{h_1} = -180°$ and $\psi_{h_m} = 180° - \Delta\psi_h$. Each data point contains the horizon pitch θ_{h_i} and horizon range R_{h_i} for a point on the horizon profile in the navigational frame $\mathcal{F}_n$ at the corresponding horizon bearing ψ_{h_i}. This horizon profile data set H^h can be defined as:

$$\mathbf{H}^h = \begin{bmatrix} -180° & -180° + \Delta\psi_h & \ldots & 180° - \Delta\psi_h \\ \theta_{h_1} & \theta_{h_2} & \ldots & \theta_{h_m} \\ R_{h_1} & R_{h_2} & \ldots & R_{h_m} \end{bmatrix} \tag{2}$$

A linear interpolating function[1] can be used on this point data set to describe the horizon profile pitch for any particular bearing ψ_h such that:

$$\theta_h = \mathbf{H}^h(\psi_h) \tag{3}$$

$$\theta_h = \theta_{h_i} + \frac{\theta_{h_{i+1}} - \theta_{h_i}}{\Delta\psi_h}(\psi_h - \psi_{h_i}) \tag{4}$$

where $\psi_{h_i} < \psi_h < \psi_{h_{i+1}}$. This interpolation function also allows the profile derivatives to be easily calculated with:

$$\frac{\partial\theta_h}{\partial\psi_h}\Big|_{\psi_h} = \frac{\theta_{h_{i+1}} - \theta_{h_i}}{\Delta\psi_h} \tag{5}$$

$$\frac{\partial\theta_h}{\partial\theta_{h_i}}\Big|_{\psi_h} = 1 - \frac{\psi_h - \psi_{h_i}}{\Delta\psi_h} \tag{6}$$

$$\frac{\partial\theta_h}{\partial\theta_{h_{i+1}}}\Big|_{\psi_h} = \frac{\psi_h - \psi_{h_i}}{\Delta\psi_h} \tag{7}$$

[1] A higher order interpolation function or spline could be used, however with a sufficiently small $\Delta\psi_h$ value the horizon shape information can be fully captured.

3.2 Generation of Horizon Profile from the Terrain Map

The horizon profile $\mathbf{H}^h\big|_{\mathbf{P}_0}$ for a specific viewing position $\mathbf{P}_0$ can be generated from a terrain map of the surrounding area. Let the reference view position expressed in the geodetic frame $\mathcal{F}_g$ be $\mathbf{P}_0^g = [\varphi_0, \lambda_0, h_0]^T$ where φ_0, λ_0 and h_0 represents the view latitude, longitude and altitude respectively. The terrain map can be from a variety of sources, however in its most basic form it usually contains terrain elevation data for a range of grid latitude and longitude locations. Let the terrain map $\mathbf{M}^g$ be a data set of n_t terrain points in geodetic frame coordinates $\mathcal{F}_g$ such that:

$$\mathbf{M}^g = \begin{bmatrix} \varphi_1 & \varphi_2 & \dots & \varphi_{n_t} \\ \lambda_1 & \lambda_2 & \dots & \lambda_{n_t} \\ h_1 & h_2 & \dots & h_{n_t} \end{bmatrix} \tag{8}$$

The first step in the generation process is to convert the terrain map $\mathbf{M}^g$ expressed in the geodetic frame $\mathcal{F}_g$ to its corresponding terrain map M^e expressed in the ECEF (Earth-Centered Earth Fixed) frame $\mathcal{F}_e$ using the point transformation:

$$\begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} = \begin{bmatrix} (N + h_j)\cos\varphi_j \cos\lambda_j \\ (N + h_j)\cos\varphi_j \sin\lambda_j \\ \left[N(1 - e^2) + h_j\right]\sin\varphi_j \end{bmatrix} \tag{9}$$

where N is the prime vertical radius of curvature and e is the eccentricity of the Earth, which are parameters that can be obtained from the WGS-84 model [33]. The terrain map data set $\mathbf{M}^e$ can be expressed relative to the view location $\mathbf{P}_0$ in the navigational frame $\mathcal{F}_n$ using:

$$\mathbf{P}_j^n = R_{ne}\left(\mathbf{P}_j^e - \mathbf{P}_0^e\right) \tag{10}$$

where $\mathbf{P}_0^e$ is the view location expressed in the ECEF frame and $\mathbf{P}_j$ is the $j \in \{1, \dots, n_t\}$ terrain point expressed in the ECEF or NED navigation frames. The transformation matrix R_{ne} to transform the ECEF frame $\mathcal{F}_e$ to the NED navigational frame $\mathcal{F}_n$ for a specific $\mathbf{P}_0^g$ view location is:

$$R_{ne} = \begin{bmatrix} -\sin\varphi_0\cos\lambda_0 & -\sin\varphi_0\sin\lambda_0 & \cos\varphi_0 \\ -\sin\lambda_0 & \cos\lambda_0 & 0 \\ -\cos\varphi_0\cos\lambda_0 & -\cos\varphi_0\sin\lambda_0 & -\sin\varphi_0 \end{bmatrix} \tag{11}$$

This terrain data set in the navigational frame can then be converted into the horizon profile data set as described in Subsection 3.1 via various methods. The simplest is to record the maximum terrain pitch for each bearing using a 3D rendering process.

3.3 Horizon Profile Transformation

A transformation method that can transform the horizon profile from the viewpoint position $\mathbf{P}_0$ to another close by viewpoint position $\mathbf{P}$ is desired for two main reasons: a) it is computationally more efficient than regenerating the horizon profile from the terrain data set for the viewpoint position $\mathbf{P}$ and b) it allows for direct evaluation of the positional derivatives of the horizon profile which can be used in the optimization process and uncertainty estimation.

It can be seen that each pair of ordered, sequential horizon points in the horizon profile forms an edge; so that the complete horizon profile describes a 3D closed polygon in which each horizon point is initially visible from the origin. If the origin of this polygon is shifted, then there is not necessarily a direct line of sight between the new origin and all of the original horizon points. The edges of the polygon can cause occlusions to block the line of sight vectors. By describing the horizon profile as a 3D horizon polygon, it allows the horizon profile to be transformed efficiently using a ray casting method. Line of sight rays are cast out from the new origin location at the required horizon profile bearings and the ray intersections are calculated to obtain the horizon profile values (θ_h, R_h) for each particular bearing (Fig.1).

Let the current horizon profile $\mathbf{H}^h\big|_{\mathbf{P}_0}$ be evaluated for the view location $\mathbf{P}_0$. The view point location is to be transformed to $\mathbf{P} = \mathbf{P}_0 + \Delta\mathbf{P}^n$ where $\Delta\mathbf{P}^n = [\Delta x, \Delta y, \Delta z]^T$ is the viewpoint positional shift expressed in the navigational frame $\mathcal{F}_n$.

The original horizon profile $\mathbf{H}^h\big|_{\mathbf{P}_0}$ can be transformed from the horizon frame $\mathcal{F}_h$ to the navigational frame $\mathcal{F}_n$ and shifted by $\Delta\mathbf{P}^n$ to form the shifted horizon polygon $\mathbf{H}^n\big|_{\mathbf{P}}$ in the navigational frame $\mathcal{F}_n$ using:

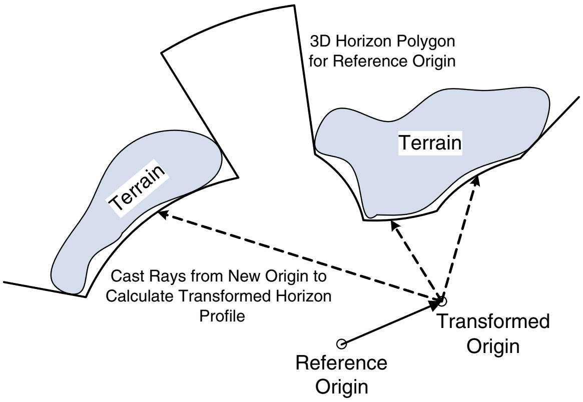

Fig. 1 Horizon polygon transformation

$$\mathbf{H}_i^n\big|_{\mathbf{P}} = \begin{bmatrix} R_{h_i}\cos\psi_{h_i} - \Delta x \\ R_{h_i}\sin\psi_{h_i} - \Delta y \\ -(R_{h_i}\tan\theta_{h_i} + \Delta z) \end{bmatrix} \tag{12}$$

A number of horizon polygon edge sets $\mathbf{E}_j\big|_{\mathbf{P}}$ can be formed from the shifted horizon polygon points to describe the complete transformed horizon polygon using:

$$\mathbf{E}_j\big|_{\mathbf{P}} = \left\{\mathbf{P}_j^n\big|_{\mathbf{P}}, \mathbf{P}_{j+1}^n\big|_{\mathbf{P}}\right\} \tag{13}$$

where $\mathbf{P}_j^n$ is the j-th point in $\mathbf{H}^n\big|_{\mathbf{P}}$, $j \in \{1, \ldots, m\}$ with m being the total number of horizon points and the horizon polygon is closed so that $\mathbf{P}_{m+1}^n\big|_{\mathbf{P}} = \mathbf{P}_1^n\big|_{\mathbf{P}}$.

Considering the single edge $\mathbf{E}_j$ shown in Fig. 2, the polygon edge line segment L_e and view ray L_r form two lines and the intersection between the two can be found. This ray casting approach allows the shifted horizon polygon to be re-sampled to form the transformed horizon profile. The two lines have the equations:

$$L_e\left(t_2\right) = \mathbf{P}_1 + t_2\left(\mathbf{P}_2 - \mathbf{P}_1\right) \tag{14}$$

$$L_r\left(t_1\right) = t_1\begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix} \tag{15}$$

and the intersection of the two lines $L_e\left(t_2\right) = L_r\left(t_1\right)$ can be found, resulting in the equations:

$$t_1 = \frac{-\mathbf{P}_{1_y}\Delta\mathbf{P}_{12_x} + \mathbf{P}_{1_x}\Delta\mathbf{P}_{12_y}}{\Delta\mathbf{P}_{12_y}\cos\psi - \Delta\mathbf{P}_{12_x}\sin\psi} \tag{16}$$

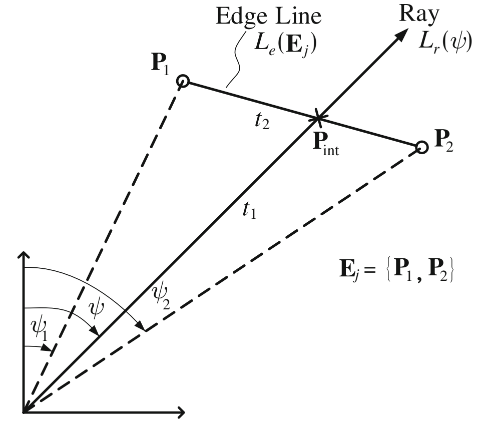

Fig. 2 Horizon edge intercept

$$t_2 = \frac{\mathbf{P}_{1_x}\sin\psi - \mathbf{P}_{1_y}\cos\psi}{\Delta\mathbf{P}_{12_y}\cos\psi - \Delta\mathbf{P}_{12_x}\sin\psi} \tag{17}$$

where $\Delta\mathbf{P}_{12} = \mathbf{P}_2 - \mathbf{P}_1$ and the conditions of $t_1 > 0$ and $0 \leq t_2 < 1$ must be satisfied for an intersection to occur. Once the intersection has been found, the horizon pitch at the intersection can be recovered using:

$$\theta_h = -\arctan\left(\frac{\mathbf{P}_{1_z} + t_2\Delta\mathbf{P}_{12_z}}{t_1}\right) \tag{18}$$

Testing every required bearing ray ψ_{h_i} with every edge $\mathbf{E}_j$ where $(i, j) \in \{1, \ldots, m\}$ is faster than generating the horizon profile from the terrain map; however this ray casting process is still inefficient. A single pass method can be developed based upon knowing the required bearing distribution. The end points of the edge segment define the angular bounds $\psi_1 \leq \psi < \psi_2$ of the segment; the segment only needs to be processed if/when the current sample ray falls into this bearing range. Using this ray tracing method, the shifted horizon polygon can be efficiently converted back into a horizon profile $\mathbf{H}^h\big|_{\mathbf{P}}$ using Algorithm 1.

The analytical jacobian $A_{\mathbf{H}}$ of this transformation process with respect to the transformed position can easily be found as part of the algorithm. This allows for computational efficiency when the transformation process is used in an optimization routine that requires derivative information. A shift in $\Delta\mathbf{P}^n$ is the same as shifting the origin of the ray casting process. This is also equivalent to negatively shifting both $\mathbf{P}_1$ and $\mathbf{P}_2$ by the same amount:

$$\frac{\partial \theta_h}{\partial \Delta \mathbf{P}^n} = -\left(\frac{\partial \theta_h}{\partial \mathbf{P}_1} + \frac{\partial \theta_h}{\partial \mathbf{P}_2}\right) \tag{19}$$

Using the relationship above, it is possible to work out the horizon pitch θ_h partial derivatives with respect to the transformed position:

$$\frac{\partial \theta_h}{\partial \Delta x} = \frac{\partial \theta_h}{\partial t_1}\frac{\partial t_1}{\partial \Delta x} + \frac{\partial \theta_h}{\partial t_2}\frac{\partial t_2}{\partial \Delta x} \tag{20}$$

$$\frac{\partial \theta_h}{\partial \Delta y} = \frac{\partial \theta_h}{\partial t_1}\frac{\partial t_1}{\partial \Delta y} + \frac{\partial \theta_h}{\partial t_2}\frac{\partial t_2}{\partial \Delta y} \tag{21}$$

$$\frac{\partial \theta_h}{\partial \Delta z} = \frac{1}{\frac{1}{t_1}\left[\mathbf{P}_{1_z} + t_2 \Delta \mathbf{P}_{12_z}\right]^2 + t_1} \tag{22}$$

where

$$\frac{\partial \theta_h}{\partial t_1} = \frac{\mathbf{P}_{1_z} + t_2 \Delta \mathbf{P}_{12_z}}{\left[\mathbf{P}_{1z} + t_2 \Delta \mathbf{P}_{12_z}\right]^2 + t_1^2} \tag{23}$$

$$\frac{\partial \theta_h}{\partial t_2} = \frac{-\Delta \mathbf{P}_{12_z}}{\frac{1}{t_1}\left[\mathbf{P}_{1_z} + t_2 \Delta \mathbf{P}_{12_z}\right]^2 + t_1} \tag{24}$$

$$\frac{\partial t_1}{\partial \Delta x} = \frac{-\Delta \mathbf{P}_{12_y}}{\cos\psi \Delta \mathbf{P}_{12_y} - \sin\psi \Delta \mathbf{P}_{12_x}} \tag{25}$$

$$\frac{\partial t_1}{\partial \Delta y} = \frac{\Delta \mathbf{P}_{12_x}}{\cos\psi \Delta \mathbf{P}_{12_y} - \sin\psi \Delta \mathbf{P}_{12_x}} \tag{26}$$

$$\frac{\partial t_2}{\partial \Delta x} = \frac{-\sin\psi}{\cos\psi \Delta \mathbf{P}_{12_y} - \sin\psi \Delta \mathbf{P}_{12_x}} \tag{27}$$

$$\frac{\partial t_2}{\partial \Delta y} = \frac{\cos\psi}{\cos\psi \Delta \mathbf{P}_{12_y} - \sin\psi \Delta \mathbf{P}_{12_x}} \tag{28}$$

These partial derivatives can be used to form the horizon pitch jacobian $A_{\mathbf{H}}$ of the transformation process:

$$A_{\mathbf{H}} = \begin{bmatrix} \frac{\partial \theta_{h_1}}{\partial \Delta x} & \frac{\partial \theta_{h_1}}{\partial \Delta y} & \frac{\partial \theta_{h_1}}{\partial \Delta z} \\ \vdots & \vdots & \vdots \\ \frac{\partial \theta_{h_m}}{\partial \Delta x} & \frac{\partial \theta_{h_m}}{\partial \Delta y} & \frac{\partial \theta_{h_m}}{\partial \Delta z} \end{bmatrix} \tag{29}$$

3.4 Image Pixel to Body Frame Horizon Vector Mapping

A mapping procedure which transforms observed horizon pixels in the image frame $\mathcal{F}_{\text{image}}$ to unit vectors in the body frame $\mathcal{F}_b$ is required so that the whole optimization cost function can operate on these unit vectors rather than the pixels in the image frame. The advantage of this is that the horizon unit vectors in the body frame are calibrated for the intrinsic camera calibration properties and rotation offsets that allow information from multiple cameras to be fused inside the optimization process without having to deal with different camera calibrations for each data set from each camera.

Let $\mathbf{p}^{\text{image}} = [u, v, 1]^T$ be a point on the horizon inside the image frame $\mathcal{F}_{\text{image}}$, where u, v are the x and y pixel coordinates in the image respectively. This point can be transformed into a direction vector $\mathbf{p}^c$ in the camera frame $\mathcal{F}_c$ and normalized using:

$$\mathbf{p}^c = K^{-1}\mathbf{p}^{\text{image}} \tag{30}$$

$$\hat{\mathbf{p}}^c = \frac{\mathbf{p}^c}{\|\mathbf{p}^c\|} \tag{31}$$

The intrinsic camera calibration matrix K is a general pinhole camera projection model [34] that maps the camera frame $\mathcal{F}_c$ to the image frame $\mathcal{F}_{\text{image}}$. This unit vector $\hat{v}_p^c$ in the camera frame $\mathcal{F}_c$ can be transformed to the body frame $\mathcal{F}_b$ using:

$$\hat{\mathbf{p}}^b = R_{bc}\hat{\mathbf{p}}^c \tag{32}$$

where R_{bc} is the inverse of the body frame $\mathcal{F}_b$ to camera frame $\mathcal{F}_c$ rotation matrix that allows for camera mounting body rotation offsets. The inverse body to camera rotation matrix R_{bc} is given by:

$$R_{bc} = R_z(-\psi_o)R_y(-\theta_o)R_x(-\phi_o) \tag{33}$$

where ϕ_o, θ_o and ψ_o are the body to camera mounting offset Euler angles.

3.5 Body Frame to Horizon Frame Vector Mapping

As part of the optimization process, a mapping algorithm is needed that takes the observed horizon vectors $\hat{\mathbf{p}}^b$ in the body frame $\mathcal{F}_b$ and transforms them via the given attitude $\mathbf{\Phi}$ to the horizon frame $\mathcal{F}_h$, so that the

Algorithm 1 Horizon Profile Generation From Horizon Polygon

```
Data: Horizon polygon edge data set $\mathbf{E}$ with the new view location at the origin.
Result: Horizon profile $\mathbf{H}^h$ evaluated for the new view location with a profile bearing
        spacing of $\Delta\psi_h = \frac{360^\circ}{m}$ and its corresponding horizon pitch positional
        jacobian $A_{\mathbf{H}} = \frac{\partial\mathbf{H}}{\partial\Delta\mathbf{P}^n}$.
begin
    for $j = 1 \rightarrow (m+1)$ do
        Obtain end point of edge segment $\mathbf{E}_j$ such that:
        $\mathbf{P}_1$ is the first point in $\mathbf{E}_j$
        $\mathbf{P}_2$ is the second point in $\mathbf{E}_j$
        Calculate the bearings of the end points using:
        $\psi_1 = \arctan\left(\frac{\mathbf{P}_{1_y}}{\mathbf{P}_{1_x}}\right)$ $\psi_2 = \arctan\left(\frac{\mathbf{P}_{2_y}}{\mathbf{P}_{2_x}}\right)$
        Calculate the bearing indices using:
        $I_{\psi_1} = \lfloor(\psi_1 + 180^\circ)/(\Delta\psi_h)\rfloor$
        $I_{\psi_2} = \lfloor(\psi_2 + 180^\circ)/(\Delta\psi_h)\rfloor$
        Correct indices for bearing discontinuity:
        if $|\psi_2 - \psi_1| \geq 180^\circ$ then
            $I_{\min} = \max(I_{\psi_1}, I_{\psi_2}) - m$
            $I_{\max} = \min(I_{\psi_1}, I_{\psi_2}) + 1$
        else
            $I_{\min} = \min(I_{\psi_1}, I_{\psi_2})$
            $I_{\max} = \max(I_{\psi_1}, I_{\psi_2}) + 1$
        end
        For each bearing within angular limits
        for $k = I_{min} \rightarrow I_{max}$ do
            Calculate current sample bearing:
            $k = k \pmod{m}$
            $\psi = k\Delta\psi_h - 180^\circ$
            Calculate intercept:
            $\Delta\mathbf{P}_{12} = \mathbf{P}_2 - \mathbf{P}_1$
            $t_1 = \frac{-\mathbf{P}_{1_y}\Delta\mathbf{P}_{12_x} + \mathbf{P}_{1_x}\Delta\mathbf{P}_{12_y}}{\Delta\mathbf{P}_{12_y}\cos\psi - \Delta\mathbf{P}_{12_x}\sin\psi}$
            $t_2 = \frac{\mathbf{P}_{1_x}\sin\psi - \mathbf{P}_{1_y}\cos\psi}{\Delta\mathbf{P}_{12_y}\cos\psi - \Delta\mathbf{P}_{12_x}\sin\psi}$
            if $t_1 > 0$ and $0 \geq t_2 < 1$ then
                Calculate $\theta_h$ and $R_h$ for bearing $\psi$:
                $R_h = t_1$
                $\theta_h = -\arctan\left(\frac{(\mathbf{P}_{1_z} + t_2\Delta\mathbf{P}_{12_z})}{R_h}\right)$
                Update $\mathbf{H}^h$ and $A_{\mathbf{H}}$ if needed:
                if $\theta_h > \theta_{h_{(k+1)}}$ then
                    $\theta_{h_{(k+1)}} = \theta_h$
                    $R_{h_{(k+1)}} = R_h$
                    $\frac{\partial\theta_h}{\partial t_1} = \frac{\mathbf{P}_{1_z} + t_2\Delta\mathbf{P}_{12_z}}{[\mathbf{P}_{1_z} + t_2\Delta\mathbf{P}_{12_z}]^2 + t_1^2}$
                    $\frac{\partial\theta_h}{\partial t_2} = \frac{-\Delta\mathbf{P}_{12_z}}{\frac{1}{t_1}[\mathbf{P}_{1_z} + t_2\Delta\mathbf{P}_{12_z}]^2 + t_1}$
                    $\frac{\partial t_1}{\partial\Delta\mathbf{P}^n_x} = \frac{-\Delta\mathbf{P}_{12_y}}{\cos\psi\Delta\mathbf{P}_{12_y} - \sin\psi\Delta\mathbf{P}_{12_x}}$
                    $\frac{\partial t_1}{\partial\Delta\mathbf{P}^n_y} = \frac{\Delta\mathbf{P}_{12_x}}{\cos\psi\Delta\mathbf{P}_{12_y} - \sin\psi\Delta\mathbf{P}_{12_x}}$
                    $\frac{\partial t_2}{\partial\Delta\mathbf{P}^n_x} = \frac{-\sin\psi}{\cos\psi\Delta\mathbf{P}_{12_y} - \sin\psi\Delta\mathbf{P}_{12_x}}$
                    $\frac{\partial t_2}{\partial\Delta\mathbf{P}^n_y} = \frac{\cos\psi}{\cos\psi\Delta\mathbf{P}_{12_y} - \sin\psi\Delta\mathbf{P}_{12_x}}$
                    $\frac{\partial\theta_{h_{(k+1)}}}{\partial\Delta\mathbf{P}^n_x} = \frac{\partial\theta_h}{\partial t_1}\frac{\partial t_1}{\partial\Delta\mathbf{P}^n_x} + \frac{\partial\theta_h}{\partial t_2}\frac{\partial t_2}{\partial\Delta\mathbf{P}^n_x}$
                    $\frac{\partial\theta_{h_{(k+1)}}}{\partial\Delta\mathbf{P}^n_y} = \frac{\partial\theta_h}{\partial t_1}\frac{\partial t_1}{\partial\Delta\mathbf{P}^n_y} + \frac{\partial\theta_h}{\partial t_2}\frac{\partial t_2}{\partial\Delta\mathbf{P}^n_y}$
                    $\frac{\partial\theta_{h_{(k+1)}}}{\partial\Delta\mathbf{P}^n_z} = \frac{1}{\frac{1}{t_1}[\mathbf{P}_{1_z} + t_2\Delta\mathbf{P}_{12_z}]^2 + t_1}$
                end
            end
        end
    end
end
```

observed horizon can be aligned to the terrain-aided horizon in the horizon frame.

Let the navigational frame $\mathcal{F}_n$ to body frame $\mathcal{F}_b$ attitude $\boldsymbol{\Phi}$ be represented as Euler angles $\boldsymbol{\Phi} = [\phi, \theta, \psi]^T$ where ϕ, θ, ψ are the bank, pitch and yaw Euler angles respectively. The rotation matrix R_{bn} can be calculated using:

$$R_{bn} = R_x(\phi) R_y(\theta) R_z(\psi) \tag{34}$$

The body frame $\mathcal{F}_b$ unit horizon vectors $\hat{\mathbf{p}}^b$ can be transformed to the navigational frame $\mathcal{F}_n$ using the inverse rotation matrix R_{nb} such that:

$$\hat{\mathbf{p}}^n = R_{bn}^{-1} \hat{\mathbf{p}}^b \tag{35}$$

Then the vectors can be transform into the horizon frame $\mathcal{F}_h$ using:

$$\psi_h = \arctan\left(\frac{\hat{\mathbf{p}}^n_y}{\hat{\mathbf{p}}^n_x}\right) \tag{36}$$

$$\theta_h = -\arcsin\left(\hat{\mathbf{p}}^n_z\right) \tag{37}$$

For simplicity, let this whole mapping process be represented as functions:

$$\psi_h = \tilde{\psi}_h\left(\boldsymbol{\Phi}, \hat{\mathbf{p}}^b\right) \tag{38}$$

$$\theta_h = \tilde{\theta}_h\left(\boldsymbol{\Phi}, \hat{\mathbf{p}}^b\right) \tag{39}$$

where $\boldsymbol{\Phi}$ is the Euler angle triplet of the platform (or the current estimate) and $\hat{\mathbf{p}}^b$ be the unit horizon vector in the body frame $\mathcal{F}_b$.

The analytical partial derivatives of this process are needed to generate the analytical jacobian for the optimization process. These derivatives are simply calculated using:

$$\frac{\partial \tilde{\psi}_h}{\partial \boldsymbol{\Phi}} = \frac{\partial \tilde{\psi}_h}{\partial \hat{\mathbf{p}}^n_x} \frac{\partial \hat{\mathbf{p}}^n_x}{\partial \boldsymbol{\Phi}} + \frac{\partial \tilde{\psi}_h}{\partial \hat{\mathbf{p}}^n_y} \frac{\partial \hat{\mathbf{p}}^n_y}{\partial \boldsymbol{\Phi}} \tag{40}$$

$$\frac{\partial \tilde{\theta}_h}{\partial \boldsymbol{\Phi}} = \frac{\partial \tilde{\theta}_h}{\partial \hat{\mathbf{p}}^n_z} \frac{\partial \hat{\mathbf{p}}^n_z}{\partial \boldsymbol{\Phi}} \tag{41}$$

where the intermediate partial derivatives are:

$$\frac{\partial \tilde{\psi}_h}{\partial \hat{\mathbf{p}}^n_x} = \frac{-\hat{\mathbf{p}}^n_y}{\left(\hat{\mathbf{p}}^n_x\right)^2 + \left(\hat{\mathbf{p}}^n_y\right)^2} \tag{42}$$

$$\frac{\partial \tilde{\psi}_h}{\partial \hat{\mathbf{p}}^n_y} = \frac{\hat{\mathbf{p}}^n_x}{\left(\hat{\mathbf{p}}^n_x\right)^2 + \left(\hat{\mathbf{p}}^n_y\right)^2} \tag{43}$$

$$\frac{\partial \tilde{\theta}_h}{\partial \hat{\mathbf{p}}^n_z} = \frac{-1}{\sqrt{1 - \left(\hat{\mathbf{p}}^n_z\right)^2}} \tag{44}$$

$$\frac{\partial \hat{\mathbf{p}}^n}{\partial \boldsymbol{\Phi}} = \frac{\partial R_{nb}}{\partial \boldsymbol{\Phi}} \hat{\mathbf{p}}^b \tag{45}$$

3.6 Horizon Profile Attitude and Position Optimization

Using information developed in the above subsections, it is now possible to describe a cost function that compares the alignment of the horizon in the body frame $\mathcal{F}_b$ to the terrain-aided horizon profile in the horizon frame $\mathcal{F}_h$. This problem can be expressed as an optimization problem where the state vector $\tilde{\mathbf{x}}$ is optimized to minimize the cost function J, so the problem can be expressed as:

$$\min_{\tilde{\mathbf{x}}} \left\| J\left(\tilde{\mathbf{x}}\right) \right\|^2 \tag{46}$$

The state vector $\tilde{\mathbf{x}}$ contains the estimated platform attitude Euler angles $\tilde{\boldsymbol{\Phi}} = [\phi, \theta, \psi]^T$ and the estimated navigational frame offset position $\widetilde{\Delta \mathbf{P}}^n = [\Delta x, \Delta y, \Delta z]^T$ so that:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{\boldsymbol{\Phi}} \\ \widetilde{\Delta \mathbf{P}}^n \end{bmatrix} \tag{47}$$

The cost function J is:

$$J = \sum_{k=1}^{n_p} w_k J_k(\tilde{\mathbf{x}}, \hat{\mathbf{p}}^b_k) \tag{48}$$

where

$$J_k(\tilde{\mathbf{x}}, \hat{\mathbf{p}}^b_k) = \mathbf{H}^h\big|_{\mathbf{P}_0 + \widetilde{\Delta \mathbf{P}}} \left(\tilde{\psi}_h(\tilde{\boldsymbol{\Phi}}, \hat{\mathbf{p}}^b_k)\right) - \tilde{\theta}_h(\tilde{\boldsymbol{\Phi}}, \hat{\mathbf{p}}^b_k) \tag{49}$$

and $\mathbf{H}^h\big|_{\mathbf{P}_0 + \widetilde{\Delta \mathbf{P}}}$ is the transformed horizon profile calculated as in Subsection 3.3 that transforms the reference horizon profile for the reference position $\mathbf{P}_0$ with the current view position offset $\widetilde{\Delta \mathbf{P}}^n$. The horizon function $\mathbf{H}^h(\psi_h)$ is the horizon profile pitch function given by Eq. 4 and the mapping functions $\tilde{\psi}_h(\Phi, \hat{\mathbf{p}}^b)$ and $\tilde{\theta}_h(\Phi, \hat{\mathbf{p}}^b)$ are the body frame $\mathcal{F}_b$ to horizon frame $\mathcal{F}_h$ mapping equations given by Eqs. 38

and 39 respectively. The minimization algorithm optimizes the cost function vector J over the complete set of detected image horizon pixels mapped to the body frame $\hat{\mathbf{p}}_k^b$ where $k \in \{1, 2, ..., n_p\}$. This process aligns the observed horizon to the terrain-aided horizon profile in the horizon frame $\mathcal{F}_h$ by minimizing the difference in the horizon pitch. To reduce optimization errors resulting from horizon detection and terrain map errors, a weighting function can be used during the optimization process. A robust m-estimator [35] can be used as the weighting function. The m-estimator "fair" weighting function [36] is:

$$w_k(J_k) = \frac{1}{1 + |J_k|/1.3998} \tag{50}$$

where J_k is the current residual.

The cost function can be minimized using a standard Gauss-Newton or Leveberg-Marquardt algorithm [37], where the optimum $\tilde{\mathbf{x}}^{(f)}$ state from the final iteration $^{(f)}$ can be used to generate attitude and position aiding measurements for fusion into an external filter. The attitude $\tilde{\Phi}$ and position $\tilde{\mathbf{P}}^e$ measurements can be calculated using:

$$\tilde{\mathbf{\Phi}} = \tilde{\mathbf{\Phi}}^{(f)} \tag{51}$$

$$\tilde{\mathbf{P}}^e = \mathbf{P}_0^e + R_{ne}^{-1} \Delta\tilde{\mathbf{P}}^{n(f)} \tag{52}$$

where $\mathbf{P}_0^e$ is the ECEF position of the reference horizon profile and R_{ne} is the ECEF frame $\mathcal{F}_e$ to navigational frame $\mathcal{F}_n$ rotation matrix evaluated at $\mathbf{P}_0^g$.

The initial condition $\tilde{\mathbf{x}}^{(0)}$ for the optimization process can be found from the current INS attitude $\tilde{\Phi}$ and position $\tilde{\mathbf{P}}^e$ estimates where:

$$\tilde{\mathbf{x}}^{(0)} = \begin{bmatrix} \tilde{\mathbf{\Phi}} \\ R_{ne}\left(\tilde{\mathbf{P}}^e - \mathbf{P}_0^e\right) \end{bmatrix} \tag{53}$$

The jacobian A_J that is required in the optimization process is:

$$A_J = \begin{bmatrix} A_{J_1} \\ \vdots \\ A_{J_{n_p}} \end{bmatrix} \tag{54}$$

The individual jacobians A_{J_k} for each measurement can be analytically evaluated using:

$$A_{J_k} = \begin{bmatrix} \frac{\partial J_k}{\partial \phi} & \frac{\partial J_k}{\partial \theta} & \frac{\partial J_k}{\partial \psi} & \frac{\partial J_k}{\partial \Delta x} & \frac{\partial J_k}{\partial \Delta y} & \frac{\partial J_k}{\partial \Delta z} \end{bmatrix} \tag{55}$$

$$\frac{\partial J_k}{\partial \phi} = \frac{\partial \theta_h}{\partial \psi_h}\frac{\partial \tilde{\psi}_h}{\partial \phi} - \frac{\partial \tilde{\theta}_h}{\partial \phi} \tag{56}$$

$$\frac{\partial J_k}{\partial \theta} = \frac{\partial \theta_h}{\partial \psi_h}\frac{\partial \tilde{\psi}_h}{\partial \theta} - \frac{\partial \tilde{\theta}_h}{\partial \theta} \tag{57}$$

$$\frac{\partial J_k}{\partial \psi} = \frac{\partial \theta_h}{\partial \psi_h}\frac{\partial \tilde{\psi}_h}{\partial \psi} - \frac{\partial \tilde{\theta}_h}{\partial \psi} \tag{58}$$

$$\frac{\partial J_k}{\partial \Delta x} = \frac{\partial \theta_h}{\partial \theta_{h_i}}\frac{\partial \theta_{h_i}}{\partial \Delta x} + \frac{\partial \theta_h}{\partial \theta_{h_{i+1}}}\frac{\partial \theta_{h_{i+1}}}{\partial \Delta x} \tag{59}$$

$$\frac{\partial J_k}{\partial \Delta y} = \frac{\partial \theta_h}{\partial \theta_{h_i}}\frac{\partial \theta_{h_i}}{\partial \Delta y} + \frac{\partial \theta_h}{\partial \theta_{h_{i+1}}}\frac{\partial \theta_{h_{i+1}}}{\partial \Delta y} \tag{60}$$

$$\frac{\partial J_k}{\partial \Delta z} = \frac{\partial \theta_h}{\partial \theta_{h_i}}\frac{\partial \theta_{h_i}}{\partial \Delta z} + \frac{\partial \theta_h}{\partial \theta_{h_{i+1}}}\frac{\partial \theta_{h_{i+1}}}{\partial \Delta z} \tag{61}$$

where $\frac{\partial \theta_h}{\partial \psi_h}$, $\frac{\partial \theta_h}{\partial \theta_{h_i}}$ and $\frac{\partial \theta_h}{\partial \theta_{h_{i+1}}}$ are given by Eqs. 5, 6 and 7 respectively in the horizon profile representation Subsection 3.1. Partial derivatives $\frac{\partial \tilde{\psi}_h}{\partial \mathbf{\Phi}}$ and $\frac{\partial \tilde{\theta}_h}{\partial \mathbf{\Phi}}$ are given by Eqs. 40 and 41 in the body frame to horizon frame mapping Subsection 3.5 and lastly, $\frac{\partial \theta_{h_i}}{\partial \Delta \boldsymbol{P}^n}$ and $\frac{\partial \theta_{h_{i+1}}}{\partial \Delta \mathbf{P}^n}$ are extracted from $A_{\mathbf{H}}$ calculated in the horizon transformation Subsection 3.3.

The jacobian A_J from the final iteration in the optimization process can be used to estimate the final attitude and position measurement uncertainty. Under the ordinary least squares assumptions, the covariance matrix $\Sigma_{\tilde{\mathbf{x}}}$ can be calculated using:

$$\Sigma_{\tilde{\mathbf{x}}} = \sigma_{\tilde{\mathbf{x}}}^2 \left(A_J^T A_J\right)^{-1} \tag{62}$$

where the solution variance $\sigma_{\tilde{\mathbf{x}}}^2$ can be estimated from the residual fit error and horizon detection measurement uncertainty such that:

$$\sigma_{\tilde{\mathbf{x}}}^2 = \frac{\mathbf{r}^T\mathbf{r}}{n_p - 6} + \sigma_{\theta_h}^2 \tag{63}$$

where $\mathbf{r}$ is the un-weighted fit residual vector (un-weighted J) and $\sigma_{\theta_h}^2$ is the horizon pitch uncertainty in the detected horizon profile from the image processing algorithm. The solution covariance matrix consists of:

$$\Sigma_{\tilde{\mathbf{x}}} = \begin{bmatrix} \Sigma_{\tilde{\mathbf{\Phi}}} & \Sigma_{\tilde{\mathbf{\Phi}}\Delta\tilde{\mathbf{P}}} \\ \Sigma_{\tilde{\mathbf{\Phi}}\Delta\tilde{\mathbf{P}}} & \Sigma_{\Delta\tilde{\mathbf{P}}} \end{bmatrix} \tag{64}$$

where $\Sigma_{\tilde{\mathbf{\Phi}}}$ can be using directly as the attitude measurement uncertainty and the position measurement uncertainty can be recovered using the transformation:

$$\Sigma_{\tilde{\mathbf{P}}^e} = R_{en}\Sigma_{\Delta\tilde{\mathbf{P}}}R_{ne} \tag{65}$$

3.7 Proposed Localization Method Overview

The complete attitude and position localization procedure to generate the aiding measurements is outlined below:

1. First, reference horizon profiles are pre-computed off-line for a series of gridded reference points which make a 3D cube above the terrain. The reference positions have a set $\Delta\varphi$ and $\Delta\lambda$ position spacing, as well as a set Δh altitude spacing. The reference profiles can be computed using the method in Subsection 3.2.
2. The first step in the real-time algorithm is to select the closest reference horizon profile $\mathbf{H}^h|_{\mathbf{P}_0}$ from the database based on the Euclidean distance between the current INS estimated platform position $\tilde{\mathbf{P}}^g$ and the reference horizon profile position $\mathbf{P}^g_0$.
3. A horizon detection algorithm is run on the current camera images and the detected horizon image pixels are extracted $\mathbf{p}_k^{\text{image}} = [u_k, v_k, 1]^T$ where $k \in \{1, 2, ..., n_p\}$.
4. The detected horizon image pixels $\mathbf{p}_k^{\text{image}}$ are mapped from the image frame $\mathcal{F}_{\text{image}}$ to the body frame $\mathcal{F}_b$ using the method in Subsection 3.4 to produce the normalized body vectors $\hat{\mathbf{p}}_k^b$ where $k \in \{1, 2, ..., n_p\}$.
5. The optimization procedure described in Subsection 3.6 is run using the detected horizon vectors $\hat{\mathbf{p}}_k^b$ and reference horizon profile $\mathbf{H}^h|_{\mathbf{P}_0}$ to estimate the current attitude $\tilde{\mathbf{\Phi}}^{(f)}$ and position $\tilde{\mathbf{P}}^{e(f)}$. The initial conditions $\tilde{\mathbf{\Phi}}^{(0)}$ and $\Delta\tilde{\mathbf{P}}^{n(0)}$ for this optimization process are calculated from the current INS estimated attitude $\tilde{\mathbf{\Phi}}$ and position $\tilde{\mathbf{P}}^g$.
6. The estimated position $\tilde{\mathbf{P}}^e$ and attitude $\tilde{\Phi}$ measurement uncertainty $\Sigma_{\tilde{\mathbf{P}}^e}$ and $\Sigma_{\tilde{\mathbf{\Phi}}}$ are calculated from the final optimization jacobian A_J. The optimized measurements and their corresponding uncertainties then can be passed to a data fusion process for INS aiding.
7. This whole real-time procedure (steps 2-7) is then re-run for the next image frames in a recursive process.

4 Simulations and Results

Results from a simulated flight sequence are presented in this section. This allows various parameters and their effects on the optimization process to be investigated and quantified.

A flight sequence of an aircraft flying along a valley near Innsbruck was simulated. The location of the simulation was selected for the extensive surrounding mountainous terrain. The attitude sequence of the simulation is shown in Fig. 3 and the flight path along with the terrain map are shown in Fig. 4. The aircraft flight path is left to right along the white line in the figure. An example terrain-aided horizon profile is shown in Fig. 5 which was generated for the position highlighted in Fig. 4 by the white circle along the simulated flight path. The horizon profile pitch and range clearly shows the effect the valley's shape has upon the profile. The terrain surrounding the aircraft obscures the distant horizon, resulting in the mountainous horizon profile.

X-plane (a flight simulation program) was used to generate the simulated imagery from a vision system mounted on the aircraft. The vision system consisted of four 90° FOV cameras mounted facing forwards, backwards, left and right, to provide a simulated 360°

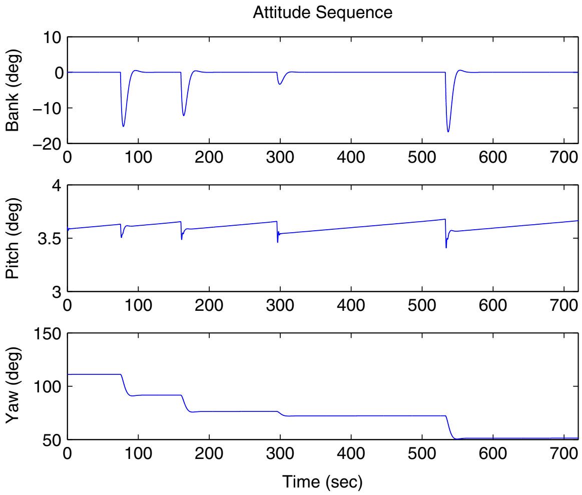

Fig. 3 Simulated flight attitude sequence

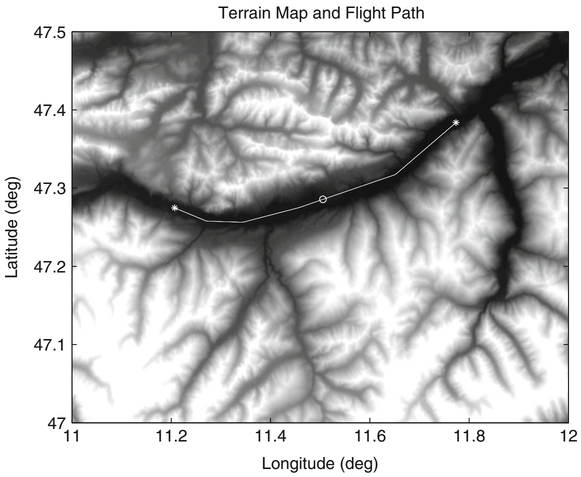

Fig. 4 Simulated flight path

view of the horizon and surrounding terrain. The camera properties for the simulated vision system are shown in Table 1. The simulated camera imagery is generated at 1 Hz. The proposed algorithm can operate at higher speeds; however the change in the horizon profile (due to the position change) at a higher frame rate can be minimal and provides little extra information for the analysis of the proposed method. It should be noted however that there is a great benefit of running the attitude estimation process at a higher rate as the rapid platform attitude response is critical for flight control.

The horizon profile detection algorithm developed in [38] was used to extract the horizon from each of the simulated camera images. The horizon detection process operates by looking for a continuous sky/ground interface edge which runs across the image. An example camera image from the simulated vision system is shown in Fig. 6 with the detected horizon profile overlayed in black. The camera image was generated for the example position along the flight path shown in Fig. 4. The horizon detection algorithm extracts an accurate representation of the visual horizon profile.

Table 1 Simulation properties

Parameter	Value
Camera Frame Rate	1Hz
Camera FOV	90°
Image Width	1024 pixels
Image Height	768 pixels
Camera 1 Attitude Offset	$[0°, 0°, 0°]^T$
Camera 2 Attitude Offset	$[0°, 0°, 90°]^T$
Camera 3 Attitude Offset	$[0°, 0°, 180°]^T$
Camera 4 Attitude Offset	$[0°, 0°, -90°]^T$

4.1 Implementation Timings

The execution timings for the different operations used in the solution are presented in Table 2. The efficiency of the proposed method is clearly visible. It would take over 3600ms to generate a single terrain-

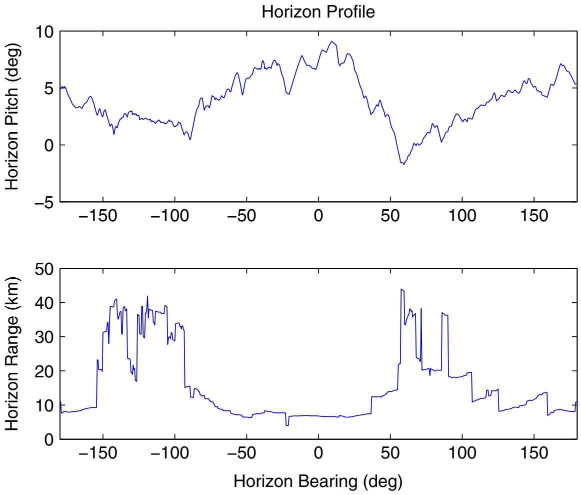

Fig. 5 Terrain-aided horizon profile

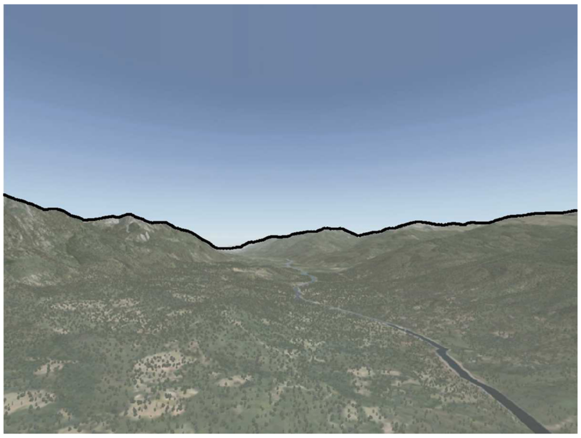

Fig. 6 Forward camera image and detected horizon profile

Table 2 Timing results

Operation	Timing[1]
Profile Generation ($n_t = 7782121$ points)	3696 ms
Profile Transformation ($m = 720$ points)	3 ms
Optimization Iteration ($n_p = 4096$ pixels)	4 ms
Complete Optimization (Mean)	53 ms
Complete Optimization (Max, 15 iterations)	73 ms
Complete Optimization (Min)	25 ms

[1] Results for an intel core 2 duo 2.33GHz processor

aided horizon profile from the terrain map.[2] However this computation time is reduced to 3ms, one thousandth of the original time by using a pre-computed reference profile and the transformation algorithm developed in Subsection 3.3. This efficient transformation process allows the optimization process to run in real-time with a mean execution time of 53ms or at 19Hz. The obtained mean execution time is a fraction of the 3 second execution time reported for the horizon based peak matching localization method developed in [10].

4.2 Storage Requirements

The computational efficiency in computational time comes at the expense of data storage requirements. A terrain database requires less storage space then the reference horizon profile database. For comparison a 1 km square terrain map sampled at 90m increments requires storage of 144 numbers. A reference horizon profile database with a grid spacing of 1km sampled in 0.5° bearings would require storage of 2880 numbers per reference altitude. This is approximately a 20 times increase in storage for a 1000 times increase in speed. It should also be noted that it is not a requirement that the reference positions are evenly spaced, so that these numbers are not the optimum.

Overall, the storage space requirements are not a major problem as storage space is cheap. It is the execution timing that is critical for aerospace guidance and control applications.

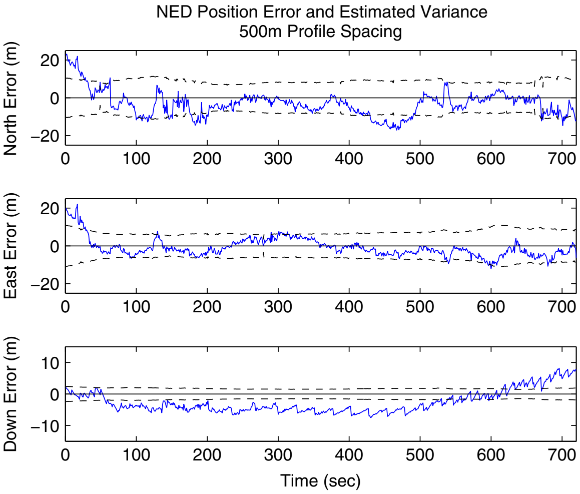

Fig. 7 NED position error with estimated 1σ bounds

4.3 Accuracy

The optimization process is run for each measurement from the simulated vision system. The errors between the optimized position and attitude measurements and the simulation truth are calculated. Initially, a database reference profile spacing[3] of 500m was used for $\Delta\varphi$ and $\Delta\lambda$ which results in the raw error graphs shown in Fig. 7 for the position error and in Fig. 8 for the attitude error. The estimated 1σ error bounds from the optimization jacobian are shown as the dashed lines in these figures. The Circular Error Probable (CEP) for the actual error sequence is shown in Fig 9.

The achieved accuracy in this simulation is highly promising. The estimated position stays within 20 meters of the truth and the estimated attitude stays tightly constrained within 0.2° of the simulated truth. This achieved accuracy for this simulation and terrain environment is higher than the typical accuracy results reported in [10, 14, 15], however a direct comparison is not possible at this stage due to the different systems and terrain used in the various papers. The estimated variance of each position measurement from the optimization process jacobian is fairly consistent with the true error. This shows that the estimated variance is a reasonable approximation of the accuracy of the measurements so it could be used in a probabilistic

[2] This time could be reduced if a GPU was used.

[3] The 500m spacing was selected as it was observed that there was insignificant increase in accuracy as the spacing decreased in this simulation.

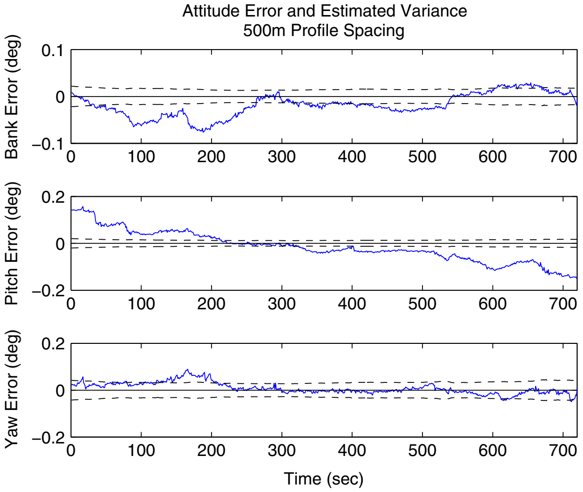

Fig. 8 Attitude error with estimated 1σ bounds

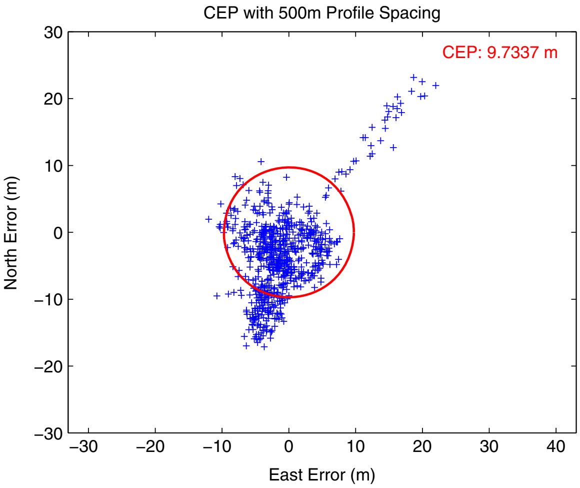

Fig. 9 Circular error probable

data fusion process as the measurement uncertainty. The error plot for the altitude show a sawtooth pattern. The time that each jump occurs correspond to the instant when the reference position updates to a closer position in the profile database.[4] This highlights how sensitive the altitude estimation process is to errors. The altitude sensitivity to pixel discretization of a single horizon pixel is approximately 10 meters. This altitude bias is not as pronounced in the results shown as it is not the altitude sensitivity to a single horizon pixel but rather it is a least squares estimate of all the horizon pixels sensitivities. This leads to future work investigating the use of aiding sensor measurements (such as from a barometer) to constrain the optimization process, removing the altitude estimation and hence sensitivities.

4.4 Horizon Detection

The pitch error graph shows a constant slope offset trend over the sequence with the pitch error sloped from positive to negative error. The CEP graph also shows a number of outlier measurements in the North-East section. Both of these erroneous results are due to errors from the horizon detection process. The optimization process is very sensitive to inconsistent data, i.e. the mismatch between the detected horizon and the terrain-aided horizon. The horizon detection error (error between the detected profile and the true terrain-aided horizon profile projected into the image) is shown in Fig. 10. From this it can be seen that there are larger horizon detection errors at the start and end of the simulation. These times correspond to when the larger attitude and position errors are reported and are a direct result of the horizon detection error. The M-estimator used in the optimization process helps to reduce the sensitivity of the solution to these horizon detection errors. The error would be an order of magnitude larger without using a robust estimation technique such as the M-estimator.

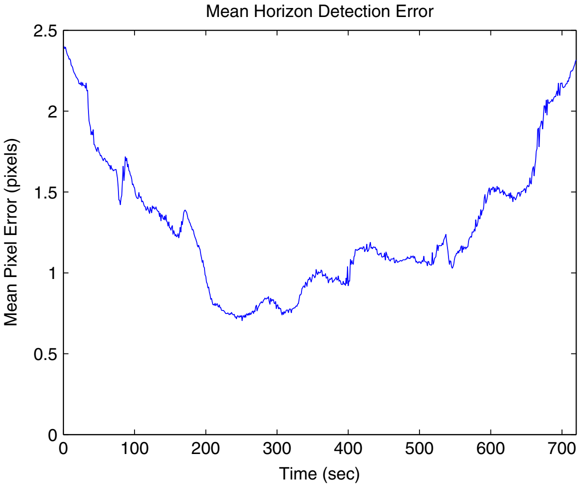

Fig. 10 Horizon detection error

[4] The sawtooth pattern is not as pronounced during the beginning of the sequence as it is a function of the local terrain and how accurately it can be estimated via the horizon profile transformation method.

The horizon detection process can only reliably pick up the visible horizon and hence the visible terrain. There can be cases when the distant terrain is washed out and undetectable due to the view distances involved or other weather effects such as haze or fog. The terrain-aided horizon profile will still contain this profile information whether it is detectable or not. This can create inconsistent data in the optimization process. This is the main cause of the horizon detection error in the simulation. At the start and end of the simulation, the backwards and then the forwards facing camera respectively are aligned along the valley facing the distant terrain (+40 km). In these situations the horizon detection algorithm cannot detect the distant horizon, so the detected profile sits lower in the image than where the distant terrain would be. This is why the pitch error is positive at the beginning of the simulation and then negative at the end of the simulation. This corresponds to the times when the horizon detection error is the greatest. The CEP accuracy for the simulation sequence is 9.76 meters, this CEP accuracy increases to 2.09 meters if the horizon detection process could accuracy select the true horizon profile each measurement.

Infra-red cameras and horizon profile detection fault detection processes could be investigated to help improve the horizon measurements and remove horizon detection errors. Additional weightings in the optimization process based on the current view distance and terrain range could be investigated to minimise errors when the distant terrain is unobservable in the image frame.

4.5 Grid Spacing

To investigate how the reference profile spacing affects the accuracy, the simulation was run multiple times with various reference profile spacings. The standard deviations of the position errors are shown Table 3 and the standard deviations of the attitude errors are shown in Table 4 for various grid densities. There is little change in the accuracy of the solution as the spacings are increased. This highlights one of the advantages of using reference profiles, most of the positional horizon information can be contained when using a fairly coarse grid. The CEP results for the various spacings are shown in Table 5. The CEP is very similar between 500 meter and 1500 meter spacings. At a 2000 meter spacing, the CEP starts to increase. The last two results shown in this table are for when the true horizon profile (rather then the one from the horizon detection process) is used. In these two cases, the CEP is quite small with little difference between 500 meter and 2000 meter results. This shows that reasonable spacings can be used and most of the error will come from the horizon detection process rather than the terrain model and transformation algorithm in the optimization process. There is also no requirement that the reference profiles need to be generated with a uniform grid spacing. There is scope for future work to use an adaptive spacing algorithm to change the grid spacing based on the local terrain shape, thus reducing the number of reference profiles required without significantly affecting the accuracy of the method.

Table 3 Reference profile spacing position error results

Spacing	North (1σ)	East (1σ)	Down (1σ)
500m	6.541m	4.877m	3.535m
1000m	6.783m	4.847m	3.712m
1500m	6.939m	5.297m	4.164m
2000m	7.637m	6.095m	4.646m

Table 4 Reference profile spacing attitude error results

Spacing	Bank (1σ)	Pitch (1σ)	Yaw (1σ)
500m	0.0247°	0.0662°	0.0240°
1000m	0.0246°	0.0663°	0.0245°
1500m	0.0255°	0.0667°	0.0269°
2000m	0.0243°	0.0683°	0.0299°

Table 5 Reference profile spacing CEP results

Profile spacing	Circular error probable
500m	9.73m
1000m	9.64m
1500m	9.94m
2000m	11.15m
500m[1]	2.09m
2000m[1]	3.07m

[1] Simulation was run using the true horizon profile in the image, thus removing any horizon detection error effects

4.6 Field of View

The use of the complete 360° FOV provides the algorithm with the best chance of observing distinct terrain that would allow the algorithm to accurate localize the platform. The complete FOV is not a prerequisite of the method, however a smaller FOV would increase the likelihood that the optimization process becomes degraded as the lateral direction translation becomes unobservable. This could be counteracted by constraining the optimization with *a priori* information or measurements from other sensors (such as a magnetometer). Future investigations need to be undertaken to determine the impact of a limited FOV.

4.7 Terrain

It is difficult to assess the impact that the shape of the terrain has on the positional accuracy of the process. It is obvious that the terrain and horizon profile require a certain level of roughness and variation so that it contains positional information. Monte-Carlo simulations were run with increasing altitude offsets to gauge the affect that the terrain shape has on the positional accuracy. As the altitude increases, the amount of terrain contained in the horizon profile reduces, until the extreme case where the horizon profile is flat and contains no positional dependance. Figure 11 shows how the horizon profile variance affects the positional accuracy of the algorithm for this test environment. As the altitude offset increases the horizon profile variance drops and the localization accuracy quickly degrades.

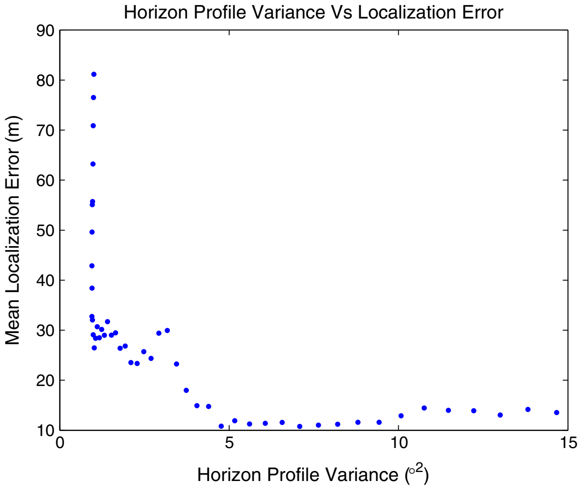

Fig. 11 Terrain variance impact on localization error

As with all terrain-aided methods, dynamically changing terrain or terrain database errors will affect the accuracy of the measurements. The robust estimator used in the optimization process helps to mitigate these effects to some extent as the whole horizon profile is used. The simulations presented here assume that the terrain is perfectly known and it is only the horizon detection process that introduces the error into process.

5 Conclusion

The numerical simulations presented show that precise real-time attitude and position measurements can be made using the information contained in the visual horizon profile. The accuracy of the proposed method still needs to be verified on flight test data with varying terrain profiles. A number of studies also need to be carried out to investigate how the amount of terrain information contained in the horizon profile will affect the positional accuracies and how a limited FOV would affect the estimation process. There is also future scope to use Infra-Red spectrum cameras to improve the horizon detection accuracy and operational range. The storage requirements of the approach could be improved by investigating a dynamic profile spacing in the reference profile database to encode the maximum amount of information without the loss of positional accuracy.

The development of a precise real-time visual horizon based localization and attitude determination method is a key step towards the development of a passive visual sensor scheme for navigation and control of small aeronautical platforms. The process alleviates the dependance on GPS for navigation which can increase the autonomy and robustness of unmanned aerial vehicles and manned flight navigation systems.

References

1. Kayton, M., Fried, W.: Avionics Navigation Systems. A Wiley-Interscience publication, Wiley (1997)
2. Bekir, E.: Introduction to Modern Navigation Systems. World Scientific (2007)

3. Titterton, D., Weston, J.: Strapdown Inertial Navigation Technology, 2nd edn. IEE Radar Series, The Institution of Engineering and Technology (2004)
4. Grewal, M., Weill, L., Andrews, A.: Global Positioning Systems, Inertial Navigation, and Integration. Wiley (2007)
5. Groves, P.: Principles of GNSS, inertial, and multi-sensor integrated navigation systems. GNSS Technology and Applications Series, Artech House (2008)
6. Farrell, J.: Aided Navigation: GPS with High Rate Sensors. McGraw-Hill professional engineering: Electronic engineering, McGraw-Hill (2008)
7. Mondragón, I.F., Campoy, P., Martinez, C., Olivares, M.: Omnidirectional vision applied to unmanned aerial vehicles (uavs) attitude and heading estimation. Robot. Auton. Syst. **58**(6), 809–819 (2010)
8. Woo, J., Son, K., Li, T., Kim, G., Kweon, I.: Vision-based UAV navigation in mountain area. In: IAPR Conference on Machine Vision Applications (2007)
9. Naval, P.: Camera pose estimation by alignment from a single mountain image. In: International Symposium on Intelligent Robotic Systems, pp. 157–163 (1998)
10. Cozman, F., Krotkov, E.: Position estimation from outdoor visual landmarks for teleoperation of lunar rovers. In: Proceedings 3rd IEEE Workshop on Applications of Computer Vision, pp. 156–161 (1996)
11. Naval, P., Mukunoki, M., Minoh, M., Ikeda, K.: Estimating camera position and orientation from geographical map and mountain image. In: 38th Research Meeting of the Pattern Sensing Group, Society of Instrument and Control Engineers, pp. 9–16 (1997)
12. Talluri, R., Aggarwal, J.: Position estimation for an autonomous mobile robot in an outdoor environment. IEEE Trans. Robot. Autom. **8**(5), 573–584 (1992)
13. Stein, F., Medioni, G.: Map-based localization using the panoramic horizon. In: IEEE International Conference on Robotics and Automation, vol. 3, pp. 2631–2637 (1992)
14. Thompson, W., Pick Jr, H.: Vision-based navigation. In: DARPA Image Understanding Workshop, pp. 149–152 (1992)
15. Thompson, W., Henderson, T., Colvin, T., Dick, L., Valiquette, C.: Vision-based localization. In: DARPA Image Understanding Workshop, pp. 491–498 (1993)
16. Baker, W., Clem, R.: Terrain contour matching [TERCOM] primer. Tech. Rep. ASP-TR-77-61, Wright-Patterson AFB Aeronautical Systems Division (1977)
17. Cowie, M., Wilkinson, N., Powlesland, R.: Latest developments of the TERPROM digital terrain system (dts). In: IEEE/ION Position, Location and Navigation Symposium, pp. 1219–1229 (2008)
18. Kubota, T., Moesl, K., Nakatani, I.: Map matching scheme for position estimation of planetary explorer in natural terrain. In: IEEE International Conference on Robotics and Automation, pp. 3520–3525 (2007)
19. Sim, D., Park, R., Kim, R., Lee, S., Kim, I.: Integrated position estimation using aerial image sequences. IEEE Trans. Pattern Anal. Mach. Intell. **24**(1), 1–18 (2002)
20. Sim, D., Park, R.: Localization based on DEM matching using multiple aerial image pairs. IEEE Trans. Image Process. **11**(1), 52–55 (2002)
21. Lerner, R., Rivlin, E., Rotstein, H.: Pose and motion recovery from feature correspondences and a digital terrain map. IEEE Trans. Pattern Anal. Mach. Intell. **28**(9), 1404–1417 (2006)
22. Lerner, R., Rivlin, E., Rotstein, P.: Error analysis for a navigation algorithm based on optical-flow and a digital terrain map. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 604–610 (2004)
23. Moe, A.: Passive aircraft altitude estimation using computer vision, thesis, Linkoping Studies in Science and Technology (2000)
24. Lee, D., Kim, Y., Bang, H.: Vision-based terrain referenced navigation for unmanned aerial vehicles using homography relationship. J. Intell. Robot. Syst. **69**, 489–497 (2013)
25. Kim, Y., Lee, D., Bang, H.: Vision-only uav navigation aided by terrain elevation map. In: International Conference on Control, Automation and Systems, pp. 1729–1733 (2012)
26. Nordlund, P.-J., Gustafsson, F.: Recursive estimation of three-dimensional aircraft position using terrain-aided positioning. In: International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. II–1121–II–1124 (2002)
27. Nordlund, P.-J., Gustafsson, F.: Marginalized particle filter for accurate and reliable terrain-aided navigation. Trans. Aerosp. Electron. Syst. **45**(4), 1385–1399 (2009)
28. Yigit, H., Yilmaz, G.: Development of a gpu accelerated terrain referenced uav localization and navigation algorithm. J. Intell. Robot. Syst. **70**(1–4), 477–489 (2013)
29. Carlson, G., Bair, G., Benoit, C.: Horizon profile checkpoints for low-altitude aircraft. IEEE Trans. Aerosp. Electron. Syst. **2**, 152–161 (1976)
30. Carlson, G., Bair, G., Benoit, C.: Geographic orientation for low-altitude aircraft using horizon matching. Tech. Rep., Defense Technical Information Center (1975)
31. Baboud, L., Cadík, M., Eisemann, E., Seidel, H.-P.: Automatic photo-to-terrain alignment for the annotation of mountain pictures. In: Conference on Computer Vision and Pattern Recognition, pp. 41–48 (2011)
32. Baatz, G., Saurer, O., Köser, K., Pollefeys, M.: Large scale visual geo-localization of images in mountainous terrain. In: Computer Vision–ECCV 2012, pp. 517–530. Springer (2012)
33. Imagery, N., Agency, M.: Department of defense world geodetic system 1984: its definition and relationships with local geodetic systems. Tech. Rep. TR8350.2, National Imagery and Mapping Agency (2000)
34. Forsyth, D., Ponce, J.: Computer Vision: A Modern Approach. Prentice Hall (2011)
35. Holland, P.W., Welsch, R.E.: Robust regression using iteratively reweighted least-squares. Commun. Stat. Theory Methods **6**(9), 813–827 (1977)
36. Rey, W.: Introduction to robust and quasi-robust statistical methods. Springer-Verlag (1983)
37. Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. **11**(2), 431–441 (1963)
38. Dumble, S., Gibbens, P.: Horizon profile detection for attitude determination. J. Intell. Robot. Syst. **68**, 339–357 (2012)