

# Autonomous Airborne Video-Aided Navigation

KYUNGSUK LEE, JASON M. KRIESEL, and NAHUM GAT  
Opto-Knowledge Systems, Inc. (OKSI), Torrance, CA 90502

*Received December 2008; Revised June 2010*

**ABSTRACT:** *We present an autonomous airborne video-aided navigation system that uses video from an onboard camera, data from an IMU, and digitally stored georeferenced landmark images. The system enables self-contained navigation in the absence of GPS. Relative position and motion are tracked by comparing simple mathematical representations of consecutive video frames. Periodically, a single image frame is compared to the landmark image to determine absolute position and correct for any possible drift or bias in calculating the relative motion. This paper describes the computational approach, test flight hardware, and test results obtained using actual flight data. The techniques are designed to be used for UAVs, cruise missiles, or smart munitions, and provide a cost effective system for navigation in GPS-denied environments.*

## INTRODUCTION

The U.S. Department of Defense relies heavily on GPS for both targeting and navigation, on soldiers, manned and unmanned ground and aerial platforms, and guided munitions. However, GPS signals are susceptible to jamming and can be difficult to utilize in certain locations, such as “urban canyons.” A high quality, low-cost backup to GPS is needed so that the targeting and navigation capabilities of the U.S. warfighter are not compromised under adverse conditions.

Inertial Navigation Systems (INS), Attitude Heading Reference Systems (AHRS), and Inertial Measurement Units (IMU) use gyros, accelerometers, and magnetometers to track motion and determine heading. While such systems can be used as GPS alternatives, the sensors are known to suffer from drift and random walk errors during long-duration operations. Position and attitude determination via double integration of accelerometers and gyros are particularly susceptible to errors. Errors accumulate over time and can produce relatively large and unacceptable mistakes in navigation. High-end INS devices take great pains to monitor the drift and bias and can do fairly well over short distances, but these systems are prohibitively expensive (cost > \$20k) for use onboard low cost platforms or expendable munitions. In addition, even these higher cost inertial devices do not have a means to make an absolute verification of

the location of the platform at any time, and must “blindly” trust the dead reckoning calculations.

Because of these limitations, reasonably priced inertial systems alone do not provide a reliable backup solution to GPS. Thus, a reliable, low cost alternative to inertial systems is needed for situations where GPS is jammed or otherwise unavailable. Such an alternative system would allow munitions and other platforms to navigate and target in GPS-denied environments, and would provide an option for lunar and planetary surface explorations where GPS is simply not available.

Our solution to this need is a video-aided navigation system (VANS) that does not rely on GPS and can work with a low cost inertial type device (and appears promising for use without an inertial system altogether). A sequence of video images contains large amounts of information that can be used for vehicle navigation and control, object detection and identification, obstacle avoidance, and many other tasks. Unlike radar- or laser-based systems, computer vision is passive and emits no external signals. As a result, vision systems can operate undetectably in hostile environments. Video camera(s), inertial device(s), batteries, etc., are generally already part of the payload of a typical Unmanned Aerial Vehicle (UAV); therefore, the additional size, weight, power, and cost requirements are minimal.

## PREVIOUS WORK

There have been many previous efforts and research projects related to computer vision based

airborne navigation systems. Most of those have focused on feature tracking and optical flow-based methods to estimate platform motion, and they compute the platform positions through the registration of images taken at multiple views (e.g., video sequence or/and landmark images). They usually utilize a variety of sensor systems to take advantage of their coupling effect or to compensate for the weakness of each system.

There have been several studies incorporating imaging systems with IMU or GPS/IMU navigation systems [1–4]. While some of the previous efforts ([1] and [2]) are limited where GPS is not available, other efforts (such as [3] and [4]) have shown the benefit of fusing imaging and inertial systems in GPS-denied environments for improved performance over inertial-only navigation systems. These efforts use a stochastic feature tracking method, employing a Kalman filter for feature correspondence searches between images. The advantage of this method is that one can track and minimize the INS-navigating error by analyzing the discrepancy between INS-predicted feature-positions and image-coregistered feature-positions where the discrepancy serves in the Kalman filter, as INS error-samples obtained from a source independent of INS itself.

Other efforts on feature tracking-based methods have been investigated based on well-known solutions for 3-D scene reconstruction [5], theories in camera calibration and image registration [6], or motion estimation [7] to provide relative platform positions.

A more accurate absolute position of a platform can also be estimated by tracking/matching the 2-D projections of located landmark features to platform sensor images [8, 9], or by reconstructing the terrain map from multiple images to compare with reference data such as Digital Elevation Models (DEMs) [10]. These techniques are computationally intensive and limited to navigation only over areas where landmark image or DEM is available.

Compared to previous techniques, our video-aided navigation technique is based on relatively simple and fast processes. Unlike other techniques that rely primarily on inertial measurements, in principle, our technique does not even require an IMU. It can work using only a sequence of video frames, an altimeter, and Digital Terrain Elevation Data (DTED) to estimate camera-pointing positions, where the camera-pointing position is the position on the ground where the extended camera optical axis is intersected. The camera-pointing position may not necessarily match the actual platform track due to the platform attitude such that the camera does not point straight down. Inertial measurements are therefore useful in determining the platform attitude changes enabling conversion

of changes in the camera pointing position to changes in ground position. The significance of the use of the INS data in the current technique is that the primary motion calculation is accomplished with the video data, not INS, and only rate values are used from the inertial system, which eliminates the need for an expensive INS.

The technique described here is divided into two basic modes of operation, Relative Navigation (RelNav) and Absolute Navigation (AbsNav). This combined approach overcomes excessive computational time, which would occur when using only absolute position determination, and overcomes potential inaccuracies when using only relative position determination.

RelNav uses video sequences to track camera-pointing position and update the current position. This algorithm is computationally fast and can execute in real time at video frame rates; however, this mode may suffer from accumulated errors over a long track due to the resolution of the cameras, the inherent distortions in video imagery, and the use of low cost IMU. Therefore, AbsNav runs periodically to update the platform's position (latitude/longitude/altitude) and attitude (i.e., roll/pitch/yaw) by correcting errors accumulated by the RelNav, and by providing an absolute reference to landmark imagery. The modes are described in the next two sections, respectively, and results obtained by applying the techniques to actual flight data are presented in the third. The techniques were initially developed under small business innovative research projects in 2002 and more specific details of how the techniques were developed as well as data from progressive tests can be found in the technical reports [11] and [12].

The goal of the work presented here is to enable the airborne platform to continue navigating without GPS by using an onboard camera, an inexpensive INS (such as an IMU), an altimeter, and an occasional comparison to landmark imagery. The techniques provide autonomous navigation capabilities for small platforms such as UAVs or Micro Air Vehicles (MAVs).

## RELATIVE NAVIGATION

The RelNav algorithm, depicted in a block diagram in Figure 1, is applied to real-time streaming video from an onboard camera. The algorithm compares successive frames in a video sequence and determines the change in the camera-pointing position from one frame to the next as illustrated in Figure 2. The algorithm uses inertial measurements to remove the perceived change in motion due to changes in the attitude of the platform as opposed to actual changes in platform position.

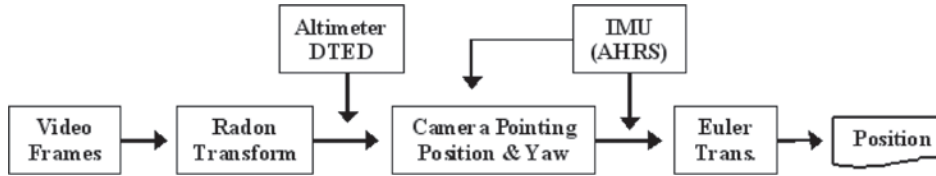


Fig. 1–Block Diagram of the Relative Navigation code.

For example, for the images in Figure 2 the platform is moving primarily in the +y direction and perceived motion in the x direction is actually due to roll motion. The image distortion between successive video frames due to the change of attitude, in principle, can be calculated using a 3D projection from ground to image plane with an Affine Transform. However, the image distortion can also be more simply approximated by rotation and translation, which is the approximation technique that RelNav uses to quickly determine the change in camera-pointing position. The change in the actual platform position is then computed with an attitude correction using IMU rate data.

### Rotation Extraction

To calculate relative rotation between frames, a Radon Transformation is applied to subsampled circular portions of the images. This circular sampling method does not suffer from the problem of different edge contents of images encountered when using rectangular images, and thus can avoid such an issue that often occurs with use of a Fourier Transform technique applied to the image data. The Radon Transform [13] is the sum of the pixels along a ray ( $s$ ) defined by the radius,  $\rho$ , from the origin, at an angle of inclination,  $\theta$ . The Radon operator maps the image domain  $I(x, y)$  to the

Radon domain, or ray-sum image,  $R(\theta, \rho)$ , in which a point corresponds to the sum of the pixels along a ray in the image domain.

$$R(\theta, \rho) = \int_{-r}^r I(\rho \cos \theta - s \sin \theta, \rho \sin \theta + s \cos \theta) ds \quad (1)$$

It is noted that two images that are rotated relative to each other produce two “ray-sum images” that are different only by a linear shift. The idea can be used to detect the rotation angle between two similar images by transforming them into the Radon domain and then extracting the shift (translation) between the two ray sum images. In practice, the ray-sum signal,  $R(\theta, 0)$ , is used instead of the full signal to save the processing time, yet still maintain comparable performance.

Figure 3(a) shows an input image (a) with rays at  $30^\circ$  equally-sampled angular directions, and Figure 3(b) shows the same image (b) rotated by  $-30^\circ$ . The ray sum array for image (a) consists of the sums ( $S1, S2, S3, S4, S5, S6$ ), while that of the rotated input image (b) consists of the sums ( $S2, S3, S4, S5, S6, S1$ ). Thus the only difference between the two ray sum signals is that the later image (b) is shifted from that of the former image (a) by “-1” translation units, where one transla-

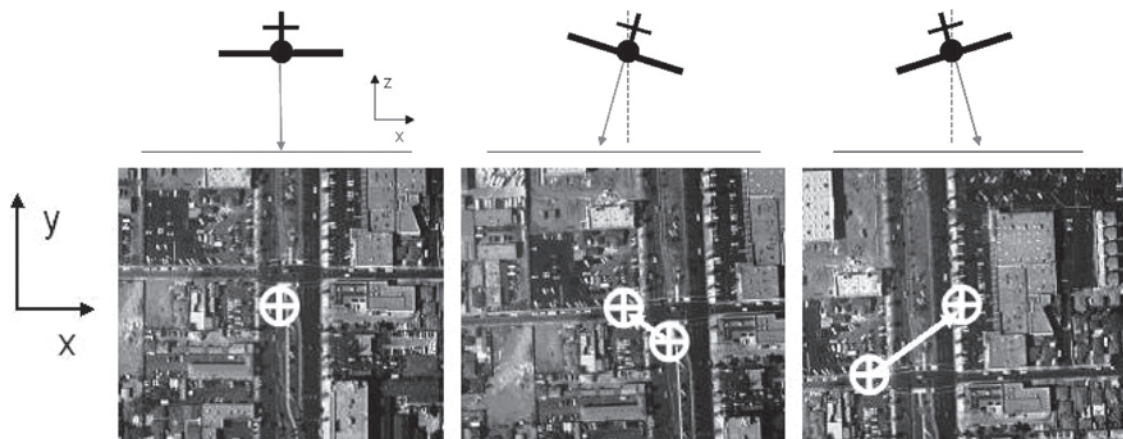


Fig. 2–Three successive frames in a video sequence showing the change in the camera pointing position from one frame to the next, along with corresponding cartoon representations of an aircraft. The change due to platform motion (+y direction) and roll motion (x direction) is calculated using image matching techniques and inertial data.

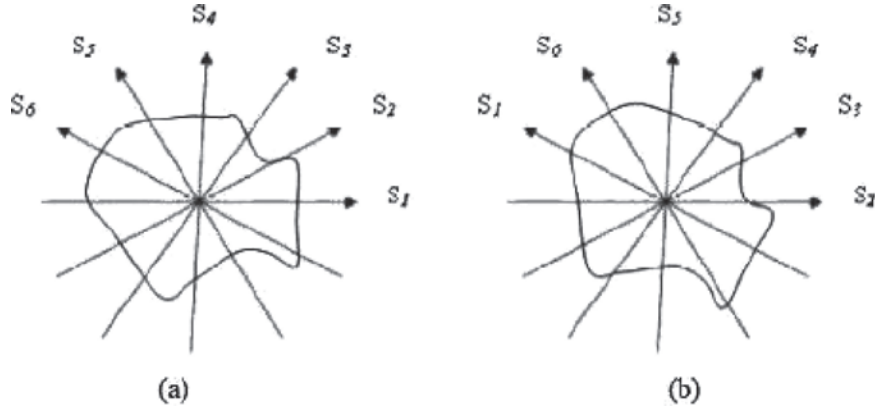


Fig. 3–Ray-sum calculation of two images of the same scene but rotated by 30°.

tion unit is equivalent to a 30° rotation. In practice, angular sampling can be done at a much higher resolution so that one translation unit would typically be 1°, 0.5°, or even less if necessary, depending on the spatial resolution of the video images. In a case where the translation unit is 1°, the ray sum signal would be a 1-D array consisting of 180 elements represented as ( $S_1, S_2, S_3, \dots, S_{180}$ ) and theoretically, the system would be able to resolve down to 1° rotation between images.

To detect the shift between the two ray sum signals, we assume that two ray sum signals,  $S(n)$  and  $T(n)$ , centered at the same point but rotated from one another, are periodic translations of one another such that  $T(n) = S(n + k)$ . Then the Fourier transforms of these two ray sum signals have the same magnitude but different phase. The relationship of Fourier transforms of  $S$  and  $T$  is described as:

$$FFT(T) = FFT(S) \cdot e^{j2\pi k u} \quad (2)$$

By the shift theorem, the difference  $k$  can be detected in terms of delta function as follows:

$$FFT^{-1} \left( \frac{FFT(S(n)) \cdot FFT^*(T(n))}{|FFT(S(n)) \cdot FFT^*(T(n))|} \right) = \delta(n - k) \quad (3)$$

In practice, it is implemented in terms of fast and simple convolution. To do this, we calculate cross-correlation scores by iterating over a given range of angles.

$$score(k) = \sum_n \frac{(S(n) - \mu_S)(T(n+k) - \mu_T)}{\sigma_S \sigma_T} \quad (4)$$

If a point in one image does not match the same point in the other image (i.e., not corresponding points), then the correlation between the two ray sum signals has a low score. The correct angular

shift between images is determined by finding the maximum cross-correlation score.

### Translation Extraction

To find the translation between frames ( $\Delta X$  and  $\Delta Y$ ) the cross-correlation score is maximized over an iterative process where the images are translated according to optimal directions. The starting position is found by calculating an estimation of the change of camera pointing position ( $\Delta \tilde{X}_c, \Delta \tilde{Y}_c$ ) using the plane velocity ( $V$ ) from a previous calculation, along with  $\Delta roll, \Delta pitch$  from the IMU measurement.

$$\Delta \tilde{X}_c = V_x \Delta t + H \cdot \tan(\Delta roll)$$

$$\Delta \tilde{Y}_c = V_y \Delta t + H \cdot \tan(\Delta pitch)$$

where  $H$  is the altitude of previous frame state.

(5)

The approximated change can easily be converted into camera pixel units as follows:

$$\left( \Delta \tilde{X}_c, \Delta \tilde{Y}_c \right)_{pixel} = \left( \Delta \tilde{X}_c, \Delta \tilde{Y}_c \right) \cdot \frac{f}{H \cdot p}$$

where  $f$  is a focal length and  $p$  is CCD pixel size.

(6)

Thus, the camera pointing position of the current frame is approximately shifted  $(\Delta \tilde{X}_c, \Delta \tilde{Y}_c)_{pixel}$  from the center position of the previous frame. Next, a Downhill Simplex method [14] is employed in an optimal search where the initial shift is used for the starting position of the Simplex run. The score that Simplex produces at a given position is the result of Radon calculation in Eq. (4), and the process is continued over a fixed number of iterations or until a sufficiently high correlation score is found. The matching point in the image that

achieves the highest score is thus the calculated change of camera-pointing position,  $(\Delta X_c, \Delta Y_c)_{pixel}$ .

### Extraction of Camera-Pointing Position and Actual Platform Position

Since the change of camera-pointing position  $(\Delta X_c, \Delta Y_c)_{pixel}$  at the current frame state is in pixel units, the pixel units are converted back into meter units to give the camera-pointing position at the current frame state as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix} \quad (8)$$

where  $\theta, \phi, \psi$  are the roll, pitch, and yaw at the current frame state. Then, using ray-tracing analysis, the vector  $(x, y, z)$  is projected onto the ground, as:

$$(x \ y \ z) \xrightarrow{\text{Ground Projection}} \left( -\frac{xH}{z} \quad -\frac{yH}{z} \quad -H \right) \quad (9)$$

Then the actual position of the platform at the current frame state is:

$$(X_p, Y_p)^{current} = (X_c, Y_c)^{current} - \left( -\frac{xH}{z}, -\frac{yH}{z} \right) \quad (10)$$

It is noted that the AGL altitude ( $H$ ) is determined from pressure altimeter data and updated at video frame rate, which provides above sea level information of the platform, together with DTED, which is a database of ground terrain elevation. Alternatively, AGL altitude can be calculated from a laser altimeter or possibly from triangulation with two platform positions. These techniques will be investigated for future research.

In summary, the RelNav continuously tracks the camera pointing positions based on frame-by-frame image analysis along with instantaneous rate values from an IMU; it then estimates actual platform position using the current platform attitude. Though the RelNav calculation utilizes inertial measurements it does not suffer in the same way from accumulated error which occurs in dead reckoning systems relying on the integration of inertial measurements over time. This is because the calculation of current platform position does not depend on previous platform position in an accumulating manner, but is calculated directly from the current camera pointing position. However, to check the position accuracy of RelNav, AbsNav

$$(X_c, Y_c)^{current} = (X_c, Y_c)^{previous} + (\Delta X_c, \Delta Y_c)_{pixel} \cdot \frac{H}{\cos(\Delta roll) \cdot \cos(\Delta pitch)} \cdot \frac{p}{f} \quad (7)$$

The actual position  $(X_p, Y_p)$  of the platform is now computed from the camera pointing position by applying a simple roll/pitch/yaw correction using Euler Transform [15] from a coordinate system centered on the platform to one centered on the ground.

is periodically invoked as described in the next section.

### ABSOLUTE NAVIGATION

The AbsNav algorithm compares a single video frame to a portion of a georeferenced landmark archival image (e.g., from a previous flight over the same area or a satellite). A difference in the spatial resolution (or GSD) between the video frame and landmark image is removed before comparison by projecting and resampling the video frame onto the same grid as the landmark image. When a match is found, the position of the platform at the time of the video frame is known. This can be used to correct the RelNav results as needed. The update rate for AbsNav algorithm depends critically on the computational resources available considering the size, weight, and power restrictions of the intended platform. A higher update rate would lead to improved accuracy, but at the cost of a higher processing burden. A typical update rate should be on the order of hundreds of seconds because the drift of the RelNav system requires it to be reset under those rates, but the optimal value depends also on the aircraft speed, the accuracy of the IMU used, and the features (or lack thereof) of the landscape.

The process of the AbsNav algorithm is broken up into two main steps: *Global Search* and *Fine Search*, as illustrated in Figure 4.

#### Global Search

The purpose of the *Global Search* is to find an approximate location (lat/lon) to be used as the starting point for a *Fine Search*. This process drastically

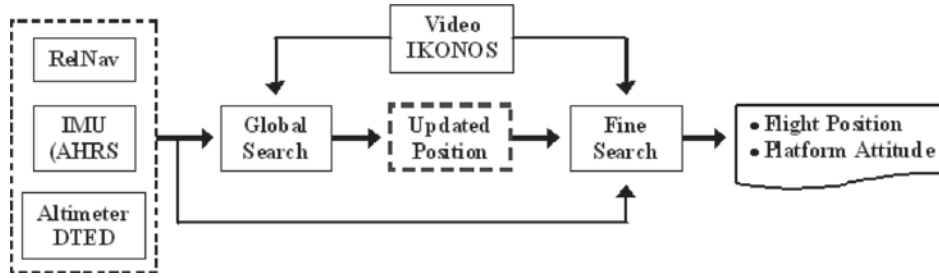


Fig. 4—Schematic of Absolute Navigation code.

saves computational time by reducing the search bound of the more detailed *Fine Search*. In the *Global Search*, the video image is projected onto the ground coordinate system to match the pixel size and orientation of the landmark image. This projection is performed based on best estimates of position (lat/lon/alt), and attitude (roll/pitch/yaw), which come from the RelNav algorithm. In order to create a projection image, we first define the Earth coordinate system (East:  $+x$ -axis, North:  $+y$ -axis) with an origin point at the lens position. The CCD image plane is then placed in an ideal position so that it is oriented identically to the Earth's  $x$ - $y$  coordinates with a nadir viewing geometry. Then the center position of the CCD plane is at  $(0, 0, f)$ , and the  $(m, n)^{\text{th}}$  pixel position in the CCD is at  $(mp, np, f)$  in the Earth coordinate system where  $p$  is the pixel size and  $f$  is the focal length of the camera lens. Given an initial set of attitude parameters and altitude (roll, pitch, yaw, and height, or  $\theta, \varphi, \psi$ , and  $h$ ), each CCD pixel vector  $(mp, np, f)$  is transformed according to the sensor's orientation using the ground projection

$$\begin{pmatrix} m & n \end{pmatrix} \xrightarrow{\text{EarthCoordinate}} \begin{pmatrix} mp & np & f \end{pmatrix} \xrightarrow{\text{EulerTransform}} \begin{pmatrix} x & y & z \end{pmatrix} \xrightarrow{\text{GroundProjection}} \begin{pmatrix} -\frac{xH}{z} & -\frac{yH}{z} & -H \end{pmatrix} \quad (11)$$

Our analysis uses a pinhole model of a camera [5]. In practice, this requires calibration of the camera to determine internal parameters such as the actual location of the image center (optical axis), lens distortions, and so on. Due to the possible errors of estimates of position (lat/lon/alt), and attitude (roll/pitch/yaw) from the RelNav, the ground projection image of the video frame usually does not exactly match to the portion of landmark image corresponding to the projected area. Around this initial area, the projected video image is “stepped” through a designated search area of the larger landmark image to find the portion of the landmark image that best matches the video image. The algorithm uses a cross correlation coefficient [16] to find a matching position.

$$\begin{aligned} r(m, n) &= \frac{\sum \sum (L(x, y) - \bar{L}(x, y)) (V(x - m, y - n) - \bar{V})}{\sqrt{\sum \sum (L(x, y) - \bar{L}(x, y))^2 \sum \sum (V(x - m, y - n) - \bar{V})^2}} \end{aligned} \quad (12)$$

where  $V$  is the video image projected onto the ground,  $\bar{V}$  is the average of  $V$ ,  $L$  is the landmark image, and  $\bar{L}$  is the average of  $L$  in the region coincident with  $V$ . The correlation coefficient is calculated along with histogram matching, where the histogram matching is used to compensate for differences in the environmental conditions, seasonal changes, and camera parameters between the present video image and the archived landmark image. The position is updated based on the best matches (highest correlation), and they are used as input to the *Fine Search* algorithm.

### Fine Search

The *Fine Search* uses the updated position and the same roll, pitch, yaw, and altitude initially used in the *Global Search*. Starting with these initial values, the *Fine Search* iterates within a local region of the landmark image until it finds an optimal set of all six degrees of freedom (roll/pitch/yaw, lat/lon/alt). For the optimizing process, we once again use the Downhill Simplex method [14]. The algorithm updates all six parameters in its optimal way, creates a rendered video image projected from the landmark image back onto the CCD plane using ray tracing (11) inversely based on the updated parameters, and finds the maximum correlation between the rendered video image and the originally captured video image. In the *Global Search*, the original video image is projected onto the landmark image (ground), while in the *Fine Search*, a portion of the landmark image is iteratively projected onto the video images while varying the projection parameters. Table 1 summarizes the two main computational modes in the AbsNav algorithm.

Table 1—Basic description of the two main computations in the *Absolute Navigation* algorithm

Process	Input	Output	Techniques
<i>Global Search</i>	<ul style="list-style-type: none"> <li>Roll/pitch from AHRS</li> <li>Altitude from DTED and altimeter</li> <li>Latitude, longitude, and yaw from <i>Relative Navigation</i></li> </ul>	<ul style="list-style-type: none"> <li>Estimated latitude and longitude (X/Y position)</li> </ul>	<ul style="list-style-type: none"> <li>Use attitude and altitude measurements to warp video image to the coordinate frame of the landmark image</li> <li>Use updated lat/lon values to set a search bound</li> <li>Step through landmark image on a global scale to find best estimate for lat/lon.</li> </ul>
<i>Fine Search</i>	<ul style="list-style-type: none"> <li>Same input as <i>Global Search</i> for roll/pitch/yaw &amp; altitude</li> <li>Updated lat/lon from <i>Global Search</i></li> </ul>	<ul style="list-style-type: none"> <li>More accurate estimate of lat/lon along with roll/ pitch/yaw &amp; altitude</li> </ul>	<ul style="list-style-type: none"> <li>Use lat/lon values from <i>Global Search</i></li> <li>Use attitude and altitude measurements as a search bound</li> <li>Iteratively warp portion of landmark with different values of attitude and altitude</li> <li>Iterate with <i>Fine Search</i> for lat/lon</li> </ul>

## RESULTS

### FLIGHT DATA COLLECTION

Flight collections for testing the video navigation system were conducted over the South Bay region of Los Angeles, California. This region has a variety of terrain providing the opportunity to collect aerial imagery over urban/suburban terrain (Torrance), undeveloped hillsides (Palos Verdes), coastline, oil refineries, industrial developments, and the Port of Los Angeles. The flight system used was comprised of various types of cameras, which were developed to fit into a standard aerial camera mount, shown in Figure 5.

Imagery was collected with several cameras, and at two different altitudes, approximately 5,000 ft and 10,000 ft above sea level. The data sets taken

as a whole can be used to optimize the algorithms for varied conditions and flight equipment, as well as to investigate potential strengths and weakness of different approaches. The cameras used for flight video capture included visible to near infrared (VNIR), a short-wave infrared (SWIR), and a long-wave infrared (LWIR) cameras to investigate different applications including the potential for day/night operations. One issue is that while thermal infrared cameras can be used at night, corresponding satellite imagery is not readily available.

The IMU system used was the Crossbow AHRS400CC [17] with nine-axis measurements, which combines linear accelerometers, rotational rate sensors, and a magnetometer. It uses an onboard DSP with Kalman filter algorithm, and has 60 Hz data rate.

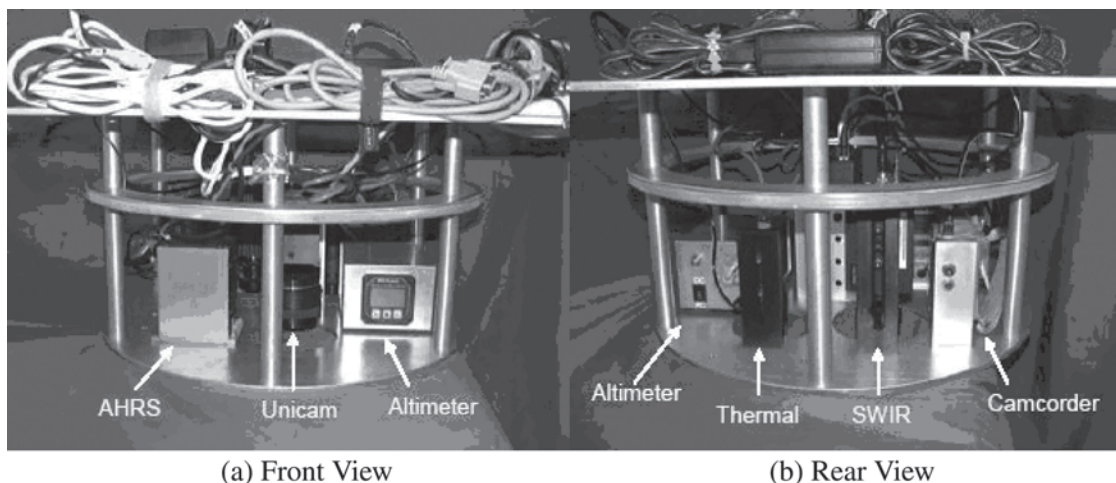


Fig. 5—Pictures of flight package used to collect aerial image data.

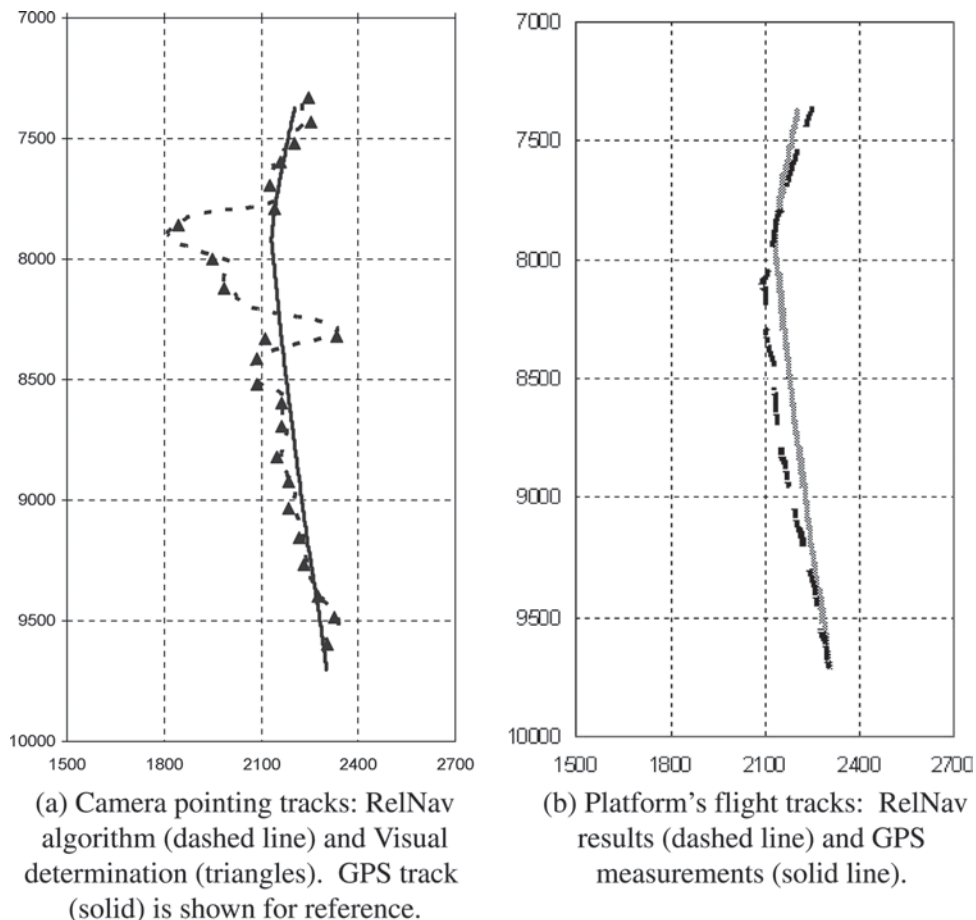


Fig. 6—Relative Navigation test results overlaid on the satellite image coordinate frame.

### TEST OF RELATIVE NAVIGATION WITH FLIGHT DATA

The RelNav algorithm was applied to the flight data, and exemplary results are shown in Figure 6. This particular test represents an extreme case as the flight line included relatively large excursions in pitch/roll/yaw around  $\pm 5^\circ/\pm 10^\circ/\pm 10^\circ$  respectively.

A comparison between calculated and reference camera-pointing tracks is shown in Figure 6(a). In the figure, the dashed line is the estimation of the camera-pointing track from the video navigation algorithm, and the triangle markers are the reference camera-pointing track obtained from a visual comparison between video image frames and satellite imagery. For the visual comparison, a set of frames was sampled from the video sequence with a gap of 50 frames between samples. Using a “human eye,” the locations of the sample frames were found in the georeferenced satellite image, and a set of pixels in the satellite image were picked, which have the best match to the center pixel of each frame. The pixels were then converted to Earth’s geo-coordinate system (lat/lon) to provide the reference camera-pointing track. This manual track provides the best test of the image

processing algorithms since the results are not dependent on potential errors associated with AHRS inaccuracies and timing issues with GPS data.

The platform’s GPS flight track and the calculated roll/pitch corrected position from the RelNav algorithm are compared in Figure 6(b) where the solid line is the platform’s flight track obtained from GPS data, and the dashed line is the estimate from the video navigation algorithm. The video navigation track is calculated by transforming the camera pointing position shown in Figure 6(a) into an aircraft position using a roll/pitch/yaw transformation based on the angles reported by the AHRS. It is noted that GPS and AHRS (IMU) data were interpolated to synchronize with the video frame rate. Figure 7 shows the error plot between the GPS flight track and the RelNav estimated track shown in Figure 6(b).

The RelNav results over this approximately 2,500-meter long flight line are summarized in Table 2, showing the RMS errors between the estimate and reference camera-pointing tracks, and between the estimate and GPS flight tracks. The RelNav position error is less than 50 m with the largest errors occurring during the most extreme maneuvers. We note that the largest source of

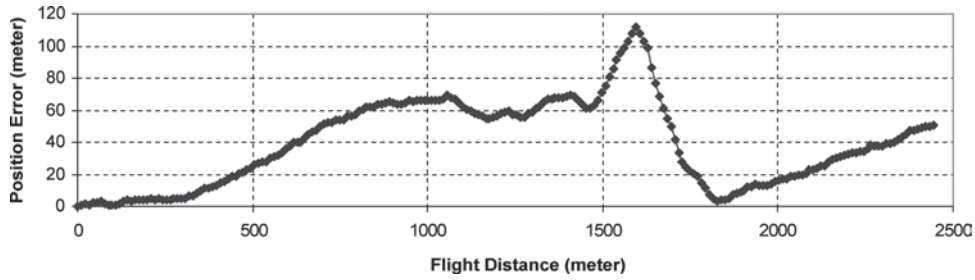


Fig. 7—Error plot of Fig. 6(b): Position errors between GPS track and estimated track of platform flight.

Table 2—RMS errors of the mismatches between estimates and references

	Camera Pointing	Flight Track
RMS	24.4 m	46.2 m

error is caused by the use of AHRS angles to convert from camera pointing position to aircraft position, and is not due to errors with the image analysis portion of the algorithm (as evident by the close agreement between calculated and reference camera pointing position shown in Figure 6(a)).

#### TEST OF ABSOLUTE NAVIGATION WITH FLIGHT DATA

The AbsNav algorithm was also tested using similar flight data. In Figure 8(a), a video frame is shown, and in Figure 8(b), the portion of the IKONOS image used for the *Global Search* is shown. This landmark image represents a relatively large search region. In practice, the location of the aircraft prior to execution

of the AbsNav code should be accurate enough that the search area can be much smaller.

At various steps in the AbsNav code, one can calculate the camera pointing position (i.e., center of the field of view) and platform location. Results of these calculations are shown in Table 3. The camera pointing position was compared to the reference camera pointing position visually determined on a  $1\text{ m} \times 1\text{ m}$  coordinate system defined by the IKONOS image. The results show that the code calculations agree very well with the visual comparison results. In other words, the algorithm correctly locates a match between an input video frame and a portion of the satellite image. In comparison, a calculation of the camera pointing position using the GPS location, along with the camera-pointing angle reported by the AHRS, differs from the actual camera pointing position by more than 200 m. Thus the combination of GPS + AHRS data do not provide an adequate reference for comparing the results of the calculation; we believe this is due to errors in the AHRS angle data on the order of  $\pm 1^\circ$ . In fact, Table 3 lists precisely how



(a) Video frame used in AbsNav test



(b) Portion of IKONOS image used in Global Search with matching portion indicated by the white rectangle

Fig. 8—Input images for AbsNav test. (a) Video image: original image is  $1,280 \times 1,024$  pixels with a GSD  $\sim 0.34$  meters. The image was resampled to  $640 \times 512$  pixels with a GSD  $\sim 2$  m for input to the algorithm. (b) Portion of IKONOS image used in Global Search; this portion is  $4,600 \times 4,000$  pixels, where each pixel is  $1\text{ m} \times 1\text{ m}$ . The image was resampled to  $2,300 \times 2,000$  pixels with a GSD  $\sim 2$  m for input to the algorithm.

Table 3—Comparison of AbsNav code outputs (*Global Search* and *Fine Search*) to measured values. Camera pointing and platform location values are in meters defined by the IKONOS coordinate grid. As discussed in the text, the “Visual Determination” provides the best test of the algorithm, to which the *Fine Search* results agree within 5 m

Case	Camera Pointing		Plane Location		Roll	Pitch	Yaw	Altitude
	X [m]	Y [m]	X [m]	Y [m]	[deg]	[deg]	[deg]	[m]
GPS & AHRS Data	2,351	1,689	2,136	1,864	-4.3	3.5	2.7	1,417
<i>Global Search</i>	2,138	1,814	1,923	1,989	-4.3	3.5	2.7	1,417
<i>Fine Search</i>	2,156	1,820	2,229	2,058	-1.6	4.8	0.9	1,404
Visual Determination	2,160	1,818	N/A	N/A	N/A	N/A	N/A	N/A

angles determined by the AbsNav algorithm differ from those reported by the AHRS. Viewed another way, the AbsNav algorithm can be used to check the accuracy of the AHRS and potentially correct for errors due to gyro bias or other sources.

Figure 9 shows a visual representation of the results presented in Table 3. Figure 9(a) shows the video frame (i.e., the same image as in Figure 8(a)). Figures 9(b) to (d) are extracted from a por-

tion of the landmark image where the center is defined in three different ways, using (b) the measured GPS and AHRS data, (c) the results of the *Global Search*, and (d) the results of the *Fine Search*. These three different images correspond to the first three rows in Table 3.

The image corresponding to the output from the *Fine Search*, Figure 9(d), is visually very similar to the actual video image taken during the flight, Fig-



(a) Unicam (Video) Image from flight taken From 4,800 ft. above sea level



(b) Image generated from IKONOS using GPS, AHRS measurements



(c) Image generated from IKONOS using Global search results and AHRS data



(d) Image generated from IKONOS using Fine search results

Fig. 9—Images related to test of AbsNav. The image shown in (a) is the flight image and the images in (b), (c), and (d) are generated from IKONOS using the values for camera pointing position and roll / pitch / yaw / altitude as listed in Table 3. The close match between (a) and (d) is a visual example of the success of the Fine Search algorithm.

ure 9(a), showing a visual demonstration of the success of the AbsNav technique. The approximate “error” in the match is less than 5 m, as shown in Table 3 (between *Fine Search* and Visual Determination). Note that there are subtle differences between the two images due to differences in the cameras used, a difference in the time of day, and actual changes to the scene from the date of the IKONOS image to the date of the flight.

One can see that the image generated using the actual GPS and AHRS measurements, Figure 9(b), is noticeably different from the actual video frame, Figure 9(a). This difference is due to a combination of inadequate time resolution of the GPS data and errors with the AHRS data (as discussed above). This difference illustrates the unreliable targeting and positioning that were obtained using the relatively inexpensive GPS and INS systems used in the test flights.

## CONCLUSION

This paper presents navigation techniques for UAVs, cruise missiles, and other platforms that use video imagery. The system does not rely on GPS and provides an autonomous image-based navigation system using inertial measurements and landmark imagery to reduce positioning error. The algorithms have been developed, tested, and optimized, and brass-board hardware has been used to collect extensive data sets for further development. The next step needed to transition the technology is the further development of a real-time system. Following this, an embedded prototype system can be produced and tested.

## ACKNOWLEDGMENTS

This work was funded by the Office of Naval Research. The authors would like to thank Mr. Joel Gat for careful reading of the manuscript and insightful comments.

## REFERENCES

1. Sullivan, D., and Brown, A., “High Accuracy Autonomous Image Georeferencing Using a GPS/Inertial-Aided Digital Imaging System,” *Proceedings of the 2002 National Technical Meeting of The Institute of Navigation*, San Diego, CA, January 2002, pp. 598–603.
2. Brown, A., Bockius, B., Johnson, B., Holland, H., and Wetlesen, D., “Flight Test Results of a Video-Aided GPS/Inertial Navigation System,” *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, Fort Worth, TX, September 2007, pp. 1111–1117.
3. Veth, M., Raquet, J., “Fusion of Low-Cost Imaging and Inertial Sensors for Navigation,” *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, Fort Worth, TX, September 2006, pp. 1093–1103.
4. Ebcin, S., and Veth, M., “Tightly-Coupled Image-Aided Inertial Navigation Using the Unscented Kalman Filter,” *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, Fort Worth, TX, September 2007, pp. 1851–1860.
5. Hartely, R. I., and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge Press, 2000.
6. Heikkila, J., and Silven, O., “A Four Step Camera Calibration Procedure with Implicit Image Correction,” *Proceedings of the Computer Vision and Pattern Recognition Conference*, San Juan, Puerto Rico, June 17–19, 1997.
7. Horn, B. K. P., Hilden, M., and Negahdaripour, S., “Closed Form Solutions of Absolute Orientation Using Orthogonal Matrices,” *JOSA-A*, Vol. 5(7), 1987.
8. Kumar, R., Sawhney, H. S., Asmuth, J. C., Pope, A., and Hue, S., “Registration of Video to Geo-Referenced Imagery,” *ICPR '98*, Brisbane, Australia, August 16–20, 1998.
9. Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q.-T., “A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry,” *Artificial Intelligence Journal*, Volume 78, 1995, pp. 87–119.
10. Rodriguez, J. J., and Aggarwal, J. K., “Matching Aerial Images to 3-D Terrain Maps,” *Pattern Analysis and Machine Intelligence*, IEEE Transactions, Vol. 12: 1990, pp. 1138–1149.
11. Gat, N., and Lee, K., “Video Based Autonomous Navigation,” Final Report: ONR SBIR Phase-I N00014-02-M-0167, November 2002.
12. Kriesel, J., Lee, K., and Gat, N., “Video Based Autonomous Navigation in GPS Denied Environments,” Final Report: ONR SBIR Phase-II N00014-03-C-0463, September 2006.
13. Deans, S. R., *The Radon Transform and Some of Its Applications*. New York: John Wiley & Sons, 1983.
14. Nelder, J. A., and Mead, R., “A Simplex Method for Function Minimization,” *Computer Journal*, Vol. 7, 1965, pp. 308–313.
15. Titterton, D. H., and Weston, J. L., *Strapdown Inertial Navigation Technology*, AIAA, 2004, Revised, 2<sup>nd</sup> edition.
16. Gonzalez, R. C., and Wintz, P., *Digital Image Processing*. Addison-Wesley, 1987, 2<sup>nd</sup> edition.
17. [http://www.xbow.com/products/product\\_pdf\\_files/inertial\\_pdf/6020-0025-01\\_b\\_ahrs400cc.pdf](http://www.xbow.com/products/product_pdf_files/inertial_pdf/6020-0025-01_b_ahrs400cc.pdf) (retrieved February 3, 2010).