Navigation in GPS Denied Environments: Feature-Aided Inertial Systems

Dr. Mikel M. Miller
Munitions Directorate
Air Force Research Laboratory
101 West Eglin Blvd, Bldg. 13, Ste 268
Eglin AFB, Florida 32542-6810
USA

Dr. Andrey Soloviev
Research and Engineering Education Facility
University of Florida
1350 N. Poquito Road
Shalimar, Florida 32579-1163
USA

Dr. Maarten Uijt de Haag
School of Electrical Engineering and Computer Science, Ohio University
321B Stocker Center, Athens, Ohio 45701
USA

Dr. Michael Veth, Dr. John Raquet
Department of Electrical and Computer Engineering
Air Force Institute of Technology
2950 P Street, Bldg. 640
Wright-Patterson AFB, Ohio 45433
USA

Dr. Timothy J. Klausutis, Dr. Jimmy E. Touma
Munitions Directorate, Air Force Research Laboratory
101 West Eglin Blvd, Bldg. 13, Ste 268
Eglin AFB, Florida 32542-6810
USA

ABSTRACT

This paper focuses on the latest technology trends for navigating in difficult urban, indoor, and underground environments where typical Global Positioning System (GPS) receivers do not function. The latest alternative navigation (Alt-Nav) technologies based on electro-optical techniques are described. These techniques extract features from electro-optical images (for example, a point feature associated with a corner of a building or a line associated with a wall of an indoor hallway) and then utilize those features as navigation related landmarks to enable aiding of inertial navigation system (INS). The Alt-Nav technologies presented include optically-aided INS and Ladar-aided INS. Tightly integrating these technologies with an INS should lead to navigation performance similar to that is achieved in today's GPS/INS integrations. An Alt-Nav integration vision for the future is given with some example configurations that improve overall navigation system robustness.

1.0 MOTIVATION AND BACKGROUND

Over the past couple of decades, there have been a number of navigation trends that have driven the desire to improve our ability to navigate in all environments. Table 1 notionally represents these trends. Previously, the primary desire was to navigate single, stand-alone systems (such as a car), but now, the desire is increasingly to have simultaneous navigation awareness of multiple interdependent systems (such as a traffic notification system in a car). Previously, navigation capability was not always counted on, but increasingly navigation is considered to be an assumed infrastructure (like knowing the lights will come on when you turn on the light switch). Previously, navigation accuracy of 5-10 m seemed almost extravagant when other worldwide navigation options prior to GPS (namely, Omega [1] and stand-alone inertial) had accuracies more on the order of 1-2 km. Now, there are many applications that require meter or sub-meter level accuracy (such as precision agriculture). Previously, due to cost, power, and size constraints, it was generally only feasible to know where the "big things" are (such as airplanes). Now, navigation is desired on more and more, smaller and smaller objects (such as cell phones).

Table 1: Navigation Trends.

Then	→	Now
Single, stand-alone systems	→	Multiple interdependent systems work together to achieve goal (requires navigation)
Precise navigation as a "nice-to-have" entity	→	Complete dependence on reliable navigation (navigation as an assumed infrastructure)
Navigation accuracy: 5-10 m is sufficient	→	Sub-meter to cm-level accuracy desired ("Accuracy is addictive")
We want to know where the "big things" are	→	We want to know where everything is

While GPS has been the driving factor behind most of these trends, there are limitations to GPS that have become more evident over time as we have increasingly come to rely on navigation. The shortfalls in GPS could be called the "navigation gap", as depicted notionally in Figure 1. The horizontal axis in this figure represents the continuum between urban/indoor and rural/open environments. The vertical axis roughly represents altitude, from ground level all the way up to space.

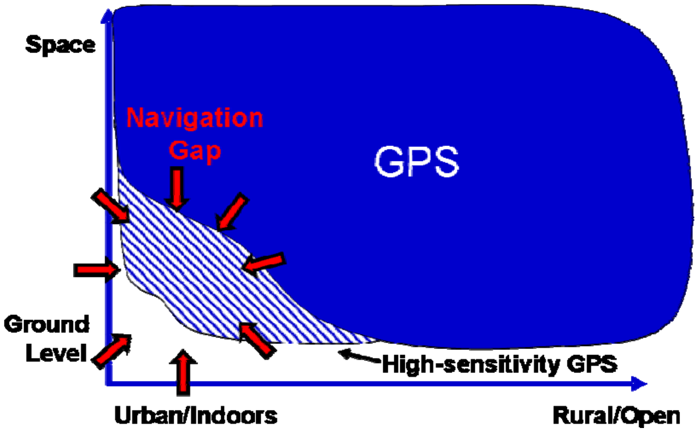


Figure 1: The Navigation Gap.

GPS does a great job of covering much of this two-dimensional trade space (indicated by the solid blue shape), but GPS by itself is not sufficient when moving close to the bottom left corner of the graph. Recent advancements in high-sensitivity GPS have helped to decrease the size of this gap (indicated by the striped blue shape), but there still remains a gap where availability, accuracy, or reliability of GPS by itself is not sufficient for many applications. Ironically, it is in just such urban/indoor locations where many people spend most of their time. (In fact, odds are that you would have a hard time obtaining a high accuracy GPS fix wherever you are reading this paper!)

Summarizing the navigation requirements in a manner similar to [2], it would be desirable to develop a navigation system that *(a) supports an indefinite mission duration, (b) supports real-time 3D location performance, (c) supports localization in urban environments and inside residential and most commercial buildings, (d) supports operation in an unknown (unmapped) or sparsely known (partially mapped) environment, (e) supports localization from the power-off condition and requires no separate starting location initialization of the user equipment, (f) supports individual isolated user terminals, (g) shall be able to re-acquire the navigation capability after a temporary loss, (h) is low-cost and low-weight, (i) does not require user motion to work, (j) shall have a level of integrity (assurance), accuracy, availability and continuity of service consistent with the tactical mission requirements.*

Alternative Navigation Techniques

For the reasons described above, alternative navigation techniques have been and are currently being developed to help fill this navigation gap. At least three broad categories of alternative navigation techniques exist:

1) **Image/ladar/Doppler/DR aiding of inertial.** These techniques attempt to use an inertial system, but constrain the drift by incorporating another source or sources of aiding. Such systems are typically self-contained. Examples include image-aided inertial navigation [3], ladar-aided inertial navigation [5], and pedometry-based DR-aiding of inertial [6]. This area of alternative navigation will be the focus of this paper.

2) **Beacon-based navigation (including pseudolites)**. If the GPS signal is not adequate for navigation in a particular environment, it is possible to transmit an additional signal or signals that are specifically designed for navigation purposes. If the transmitted signals are similar to GPS signals, then such beacon transmitters are usually called “pseudolites.” Examples of beacon-based navigation systems for indoor navigation can be found in [7] and [8].

3) **Navigation using signals of opportunity (SoOP)**. Signals of opportunity, as defined in this paper, are radio frequency (RF) signals that are not intended for navigation. Examples from previous research include digital television [9], analog television [10], and AM radio [11][12]. Part 2 of this paper series focused on these techniques [13].

This paper is focused on the first category listed above—navigation techniques that use Electro-optical sensors such as vision cameras and Ladar (laser radar or laser detection and ranging) sensors to aid an Inertial Measurement Unit (IMU). The basic idea behind these techniques is to extract features from electro-optical images (for example, a point feature associated with a corner of a building) and apply those features for landmark-based inertial aiding. Even though we focus on vision-aided and Ladar-aided navigation in this paper, it does not imply that the other alternative navigation approaches are inferior. There are strengths and weaknesses to each approach, and selecting the appropriate approach requires knowledge of the constraints and requirements of a specific application.

2.0 IMAGE-AIDED INS

Image-aided navigation serves as an excellent example of passive signal-of-opportunity navigation for the following reasons. First, there is a strong natural precedent in the animal kingdom. Many animals have been shown to utilize visual information for navigation. In fact, the ocular-vestibular system provides the primary navigation suite for humans. Secondly, optical sensors are inherently high-bandwidth. This results in the potential for very precise angular resolution. Finally, digital imaging sensors are readily available and easy to interface with, which makes them a very practical solution for investigation of navigation potential. Parts of this section are based on work presented in more detail in [3] and [4].

Image-aiding methods are typically classified as either feature-based or optic flow-based, depending on how the image correspondence problem is addressed. Feature-based methods determine correspondence of features (or "landmarks") in the scene over multiple frames, while optic flow-based methods typically determine correspondence for a whole portion of the image between frames. Optic flow methods have been proposed generally for elementary motion detection, focusing on determining relative velocity, angular rates, or obstacle avoidance [14]. Feature tracking-based navigation methods have been proposed both for fixed-mount imaging sensors or gimbal-mounted detectors which "stare" at the target of interest, in a manner similar to the gimballed infrared seeker on heat-seeking, air-to-air missiles. Many feature tracking-based navigation methods exploit knowledge (either *a priori*, through binocular stereopsis, or by exploiting terrain homography) of the target location and solve the inverse trajectory projection problem [15][16]. If no a priori knowledge of the scene is provided, estimation of the navigation state is completely correlated with estimating the scene. This is referred to as the structure from motion (SFM) problem. A theoretical development of the geometry of fixed-target tracking, with no a priori knowledge is provided in [17]. An online (extended Kalman filter-based) method for calculating a trajectory by tracking features at an unknown location on Earth's surface, provided the topography is known, is given in [18].

2.1 Basic Concept

The basic concept of the feature-based vision/inertial integrated navigation is illustrated in Figure 2.

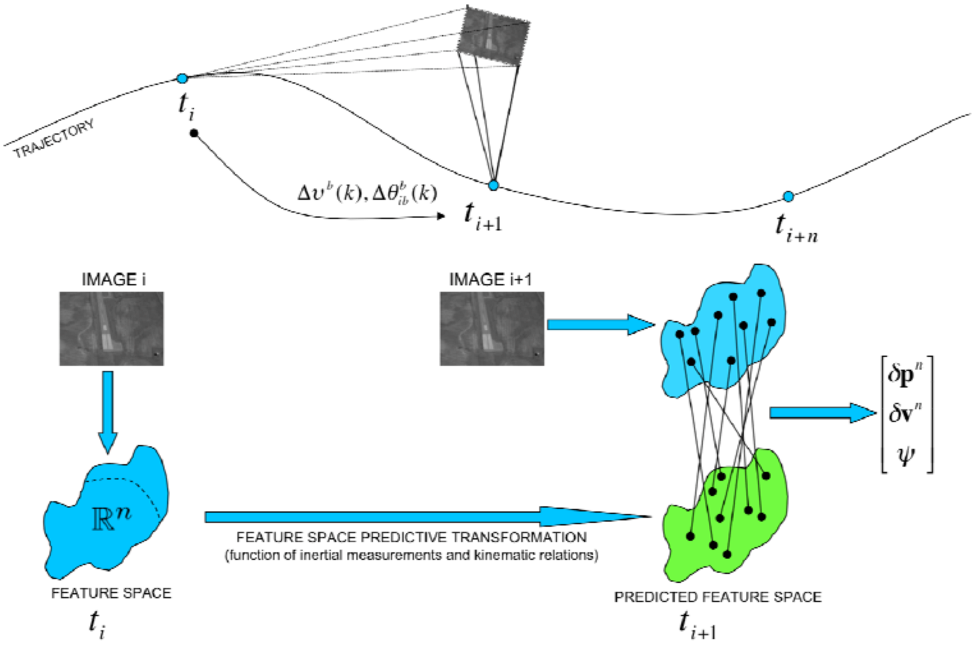

TRAJECTORY
t_i
$\Delta v^b(k), \Delta\theta^b_{ib}(k)$
t_{i+1}
t_{i+n}
IMAGE i
IMAGE i+1
$\mathbb{R}^n$
FEATURE SPACE PREDICTIVE TRANSFORMATION
(function of inertial measurements and kinematic relations)
$\begin{bmatrix} \delta\mathbf{p}^n \\ \delta\mathbf{v}^n \\ \psi \end{bmatrix}$
FEATURE SPACE
t_i
PREDICTED FEATURE SPACE
t_{i+1}

Figure 2: Overview of the image-aided inertial algorithm [4].

The algorithm consists of the following fundamental steps: At time epoch t_i the system captures an image of the environment using its digital imaging device and converts this image to a set of discrete features expressed in a space referred to as the feature space. Next, both the navigation state and the feature space state are propagated to time epoch t_{i+1}, the next imaging event. At t_{i+1} another image is captured and transformed to feature space. Both the feature space of the captured image frame at time t_{i+1} and the propagated feature space are input to the statistical feature correspondence algorithm that associates features at time t_i to features at time t_{i+1}. Finally, the trajectory error is estimated using these associated features in a Kalman estimator. The next couple of paragraphs will explain these steps in more detail.

A digital imaging device (the vision camera) measures the light intensity pattern projected through optics onto a sensor. The projection is a nonlinear function of the scene and lighting conditions, the camera optics, and the pose of the camera relative to the scene. The measured image can be thought of as a surface in three dimensions (two spatial and one intensity), which is corrupted by measurement noise, optical distortions, and possibly spatial aliasing. While humans easily interpret camera images, an automated navigation algorithm would require an automated image pre-processing step that consists of the extraction of distinct features (or landmarks in terrain referenced navigation (TRN) applications) in the image. Since the features are represented in their own space, the so-called “feature space”, the process is often referred to as the transformation of the image frame to feature space. In summary, the feature space transformation converts the image array into a collection of feature vectors, located in the feature space.

Many potential feature transformation methods have been proposed. The most desirable feature space transformation for image-aided navigation decomposes the feature space into separable pose and object dimensions. An ideal set of pose dimensions would be completely dependent on the relative pose of the object, and independent on the type of object or feature. Conversely, the ideal set of object dimensions would be completely independent of the relative pose of the object, and only depend on the type of object. This is similar to human interpretation of objects. For example, to our eyes a pencil always looks like a pencil, regardless its orientation or size. Although no algorithm is currently known which can achieve true independence between the pose and object bases, feature transformation algorithms have been proposed which display scale and rotation independence. An example of such a transformation is based on the scale-invariant feature-tracking (SIFT) algorithm developed by Lowe [19][4].

Once a collection of features is identified for the current image at time t_i, the estimated changes in the navigation state are used to predict the location of features at time t_{i+1} through a stochastic projection transformation. This transformation operates on the pose dimensions of each feature vector from the previous image and uses IMU measurements in a Kalman filter to predict pose and pose uncertainty at the time of the next image. The concept is illustrated in Figure 3.

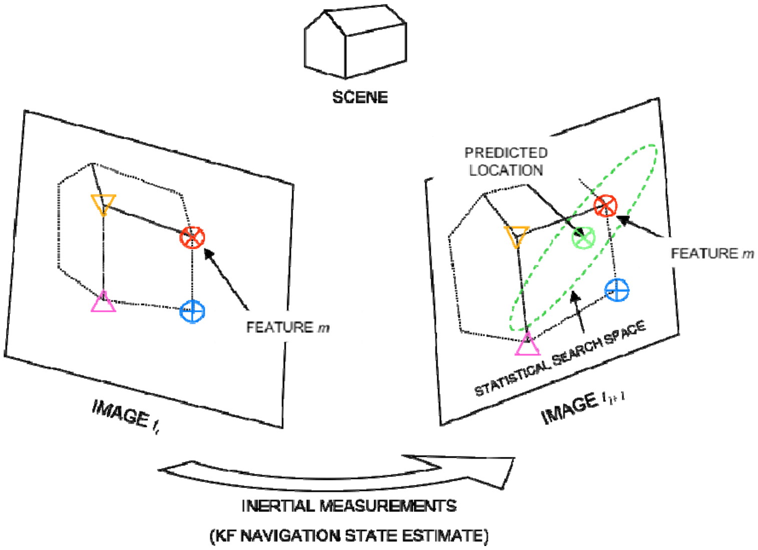


Figure 3: Stochastic feature projection. Optical features of interest are mapped into future images using inertial measurements and stochastic projections [3].

This predicted set of feature vectors is then compared to the features detected in the next image (t_{i+1}). This comparison or matching procedure compares the respective feature vectors using a rigorous, statistical weighting. This rigorous statistical feature matching method is based on inertial measurements and integrates the image and inertial sensors at a deeper level than previous methods. Once feature matches are determined, the errors between the predicted feature location and the actual feature location are used to correct errors in the navigation state. This is accomplished through the use of a Kalman filter. The Kalman filter mechanization will be described in more detail in Section 2.2. The cycle is then repeated for each image captured. In the next section, a detailed discussion of the components of the image and inertial fusion algorithm is presented.

2.2 Mechanization

Fusion of image and inertial data is accomplished through the implementation of an Extended Kalman Filter (EKF) [20][21]. The EKF estimates the errors in the calculated system parameters. In order to minimize the effects of linearization errors, the system parameters are periodically corrected by removing the current error estimate. A block diagram of the image-aided navigator is shown in Figure 4. To enable derivation of the EKF a relationship between the basic measurement (the pixel location of the feature in the image plane) and the state (position and velocity of the user) must be established. This relationship is non-linear and a general expression is given by:

$$\tilde{\mathbf{z}}(t_i) = \mathbf{h}\left\{\mathbf{p}_N(t_i), \mathbf{C}_b^N(t_i), \mathbf{y}_N(t_i), \mathbf{T}_c^{pix}\right\} + \boldsymbol{\nu}(t_i) \qquad (1)$$

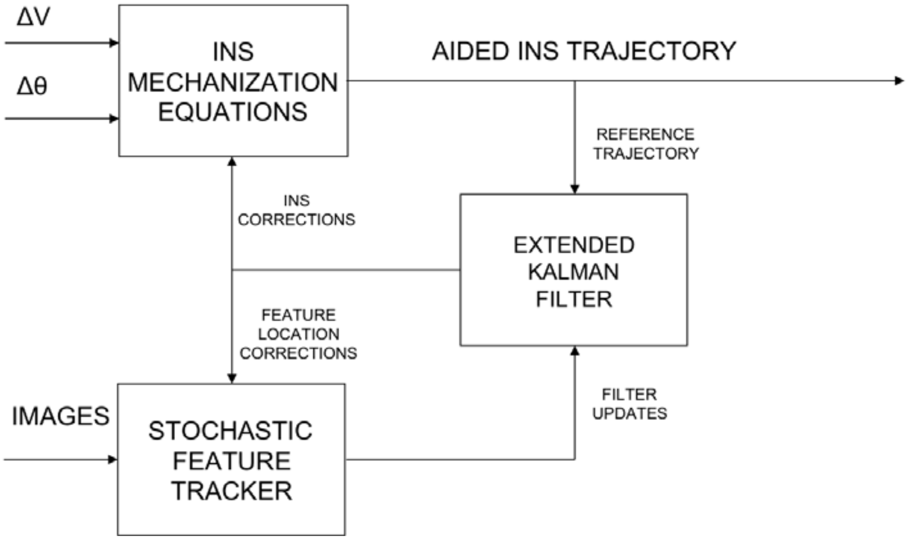


Figure 4: Image-aided inertial navigation filter block diagram with EKF.

In (1), $\mathbf{p}_N$ is the user position in the navigation frame, $\mathbf{C}_b^N$ is the body to navigation frame direction cosine matrix (DCM), $\mathbf{y}_N$ is the feature location in the navigation frame, $\mathbf{T}_c^{pix}$ is the homogeneous camera projection matrix, $\boldsymbol{\nu}$ is additive white Gaussian noise (AWGN), and $\mathbf{h}$ is a non-linear function.

Similar to equation (1), a non-linear relationship exists between the feature location in the navigation frame and the feature location measurements output by the feature extraction:

$$\tilde{\mathbf{y}}_N(t_i) = \mathbf{g}\left\{\tilde{\mathbf{p}}_N(t_i), \tilde{\mathbf{C}}_b^N(t_i), \tilde{\mathbf{z}}(t_i), \tilde{d}(t_i), \mathbf{C}_c^b, \mathbf{T}_c^{pix}\right\} \qquad (2)$$

where d is the distance form the camera to the landmark and $\mathbf{C}_c^b$ is the camera to body DCM. In equations (1) and (2) the tilde indicates that these quantities include an error component.

In the EKF implementation described in [4], the navigation error state vector, δ**x**, consists of the following fifteen elements:

$$\delta\mathbf{x} = \begin{bmatrix}\delta\mathbf{p}_N^T & \delta\mathbf{v}_N^T & \boldsymbol{\psi}^T & \mathbf{a}_b^T & \mathbf{b}_b^T\end{bmatrix}^T \qquad (3)$$

where $\delta\mathbf{p}_N$ is the position error state expressed in the navigation frame (3x1 vector), $\delta\mathbf{v}_N$ is the velocity error state expressed in the navigation frame (3x1 vector), $\boldsymbol{\psi}$ is the attitude (3x1 vector), $\mathbf{a}_b$ is the accelerometer bias expressed in the body frame (3x1 vector), and $\mathbf{b}_b$ is the gyroscope bias expressed in the body frame (3x1 vector). Feature locations are incorporated in the Kalman filter by augmenting the state vector in equation (3) with a vector containing the feature (landmark) errors:

$$\delta\mathbf{y} = [\underbrace{\delta\mathbf{y}_{N,1}^{T} \quad \cdots \quad \delta\mathbf{y}_{N,M}^{T}}_{\text{Features}}]^{T} \tag{4}$$

This addition enables tracking of these features over time and using the features for IMU error calibration. The linearized measurement model required for the EKF implementation can now be derived from equation (1) by linearizing the non-linear function **h** around the nominal trajectory by finding the derivatives of **h** with respect to the state vector elements **x** and feature elements **y**, and evaluating these at the nominal trajectories $\bar{\mathbf{x}}$, and $\bar{\mathbf{y}}$:

$$\delta\mathbf{z}_{z,i}(t_{i+1}) = \mathbf{H}_{zx}\delta\mathbf{x}(t_{i+1}) + \mathbf{H}_{zy_{n,i}}\delta\mathbf{y}_{n,i}(t_{i+1}) + \boldsymbol{\nu}(t_i) \tag{5}$$

where $\mathbf{H}_{zx}$ and $\mathbf{H}_{zy}$ are referred to as the influence or observation matrices.

Incorporating landmarks or features of opportunity into the navigation filter requires two steps: 1) determining the initial feature location and associated influence matrices ($\mathbf{H}_{zy}$), and 2) calculating the measurement errors and associated influence matrices ($\mathbf{H}_{zy}$). Various methods exist to determine the initial feature location and associated influence matrices. The methods differ in the manner in which they estimate the distance, d, from the camera(s) to the feature used in equation (2). Reference [4] addresses three methods in detail: a) feature/landmark location estimation using a statistical terrain model (one camera and a map), (b) a feature location estimation using binocular stereopsis (two cameras), (c) and a feature location estimation using egomotion and monocular measurements (one camera over time).

To complete the EKF, a state equation is required as well. This state equation or system model is characterized by state transition matrix **F**. The EKF implementation is a discrete-time implementation of the continuous-time first order Gauss-Markov state space equation:

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{6}$$

The state-transition matrix **F** and its entries can be found in [4], and for the derivation of its discrete-time version, **Φ**, the reader is referred to references [4] and [20].

In the EKF implementation, the state-transition matrix **Φ** is used to project both the navigation error state ($\delta\mathbf{x}$) and the feature error state ($\delta\mathbf{y}$) and the covariances of these error states from one time epoch to the next epoch, resulting in an navigation and feature error prediction. Next, the observations, the influence matrix, and the prediction covariances are used to calculate the Kalman gain and update the prediction to obtain the error estimate for time epoch t_{i+1}. This iterative process of projecting and updating the state vector is illustrated in Figure 5.8 of [21].

2.3 Performance Demonstration

This section will show some test results of the algorithms discussed in the previous section including simulation results and results from actual operation in a real world environment. The environment in both scenarios was an indoor hallway environment. The simulated hallway environment was simulated to be straight and 3m-by-3m-by-100m in size with randomly spaced features on walls, floors, and ceilings (0.25

features / m^2). The motion profile consisted of a 10-minute straight trajectory. Figure 5 shows the results for the horizontal error.

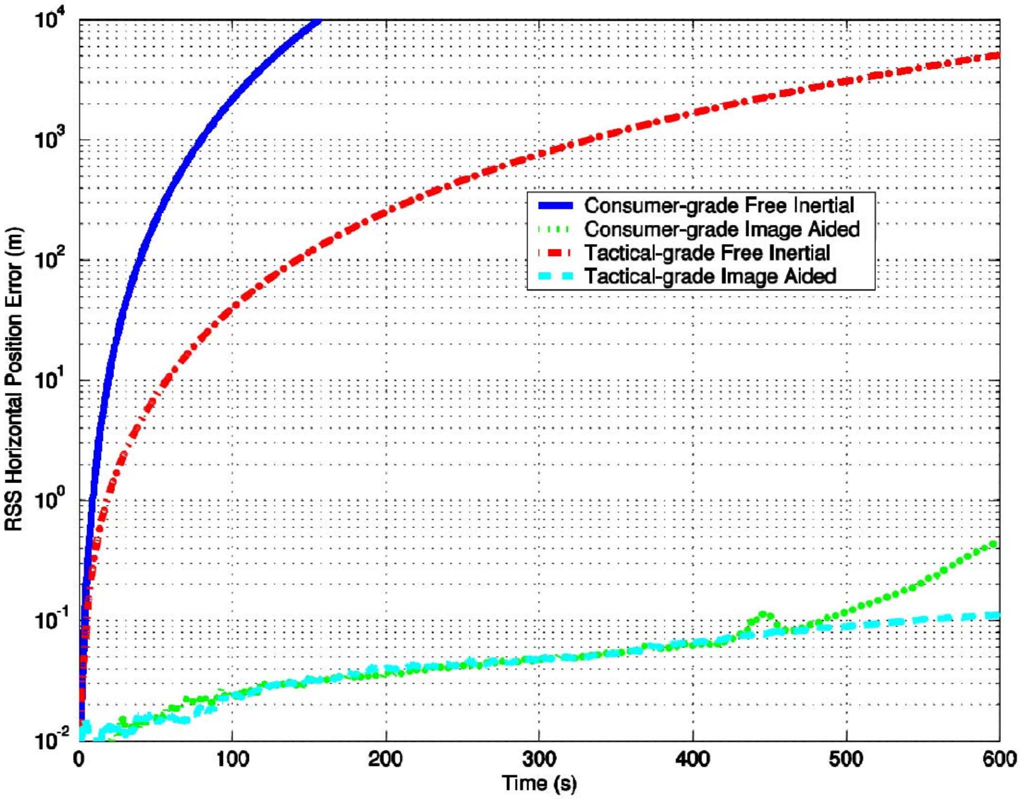


Figure 5: Simulated 60-run Monte Carlo root-sum-squared (RSS) horizontal position error for indoor profile using both consumer-grade and tactical-grade inertial sensors.

Figure 6 shows the results for some real indoor test data using both tactical and navigation-grade sensors. The profile consisted of a closed path in an indoor environment. The path began and ended at the same location and orientation in the Advanced Navigation Technology (ANT) Center laboratory, at the Air Force Institute of Technology. As in the previous profile, the data collection began with a 10 min stationary alignment period, followed by a 10 min loop around the hallways of the building. The sensor was pointed primarily in the direction of travel. No prior knowledge was provided to the algorithm regarding the location of features or structure of the environment. A sample image from the indoor profile is shown in Figure 7. The indoor profile presents the algorithm with various challenges from a feature tracking perspective. The repetitive, visually identical features (e.g., floor tiles, lights, etc.) present in the indoor environment at times confuse the feature-tracking algorithm. In addition, reflections from windows and other shiny surfaces are not always properly interpreted the filter resulting in the occasional navigation errors. Finally, the lower light intensity levels and large areas with poor contrast (e.g., smooth, featureless walls) present a relatively stark feature space.

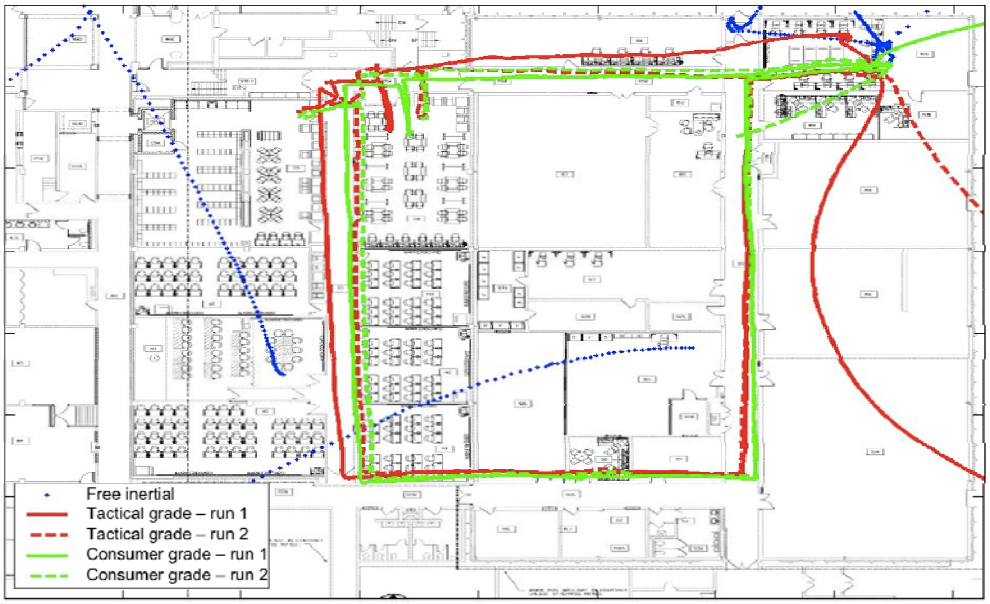


Figure 6: Results of the image-aided inertial navigator using tactical and navigation grade IMUs.

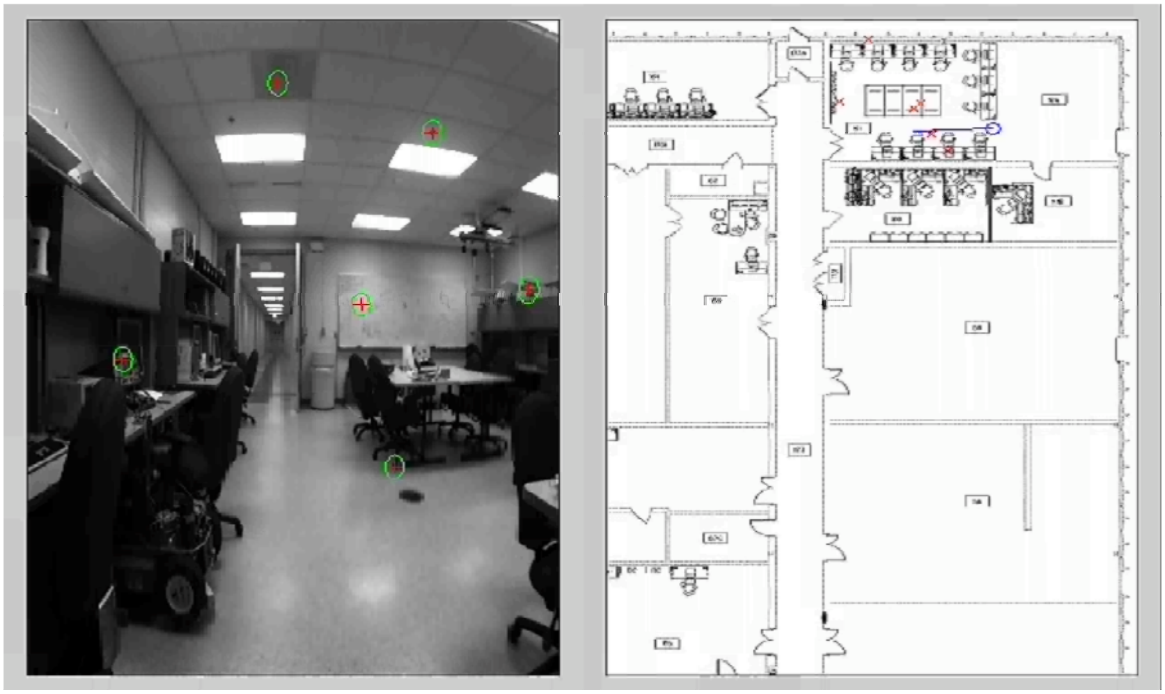


Figure 7: Example of the feature-tracking algorithm in an indoor environment.

2.4 Multi-Aperture Vision Aiding

This section discusses extension of the vision-aided inertial navigation approach for multi-aperture camera cases. Inspired from biological systems, a multi-aperture vision processing system allows for accurate motion estimation by observing optical flow across all apertures. The multi-aperture approach is particularly well suited for resolving the motion scale-factor ambiguity (i.e., for the estimation of ranges to vision-based features) by providing a wide filed of regard for detecting and tracking visual features. Figure 8 illustrates an example implementation of the multi-aperture camera system.

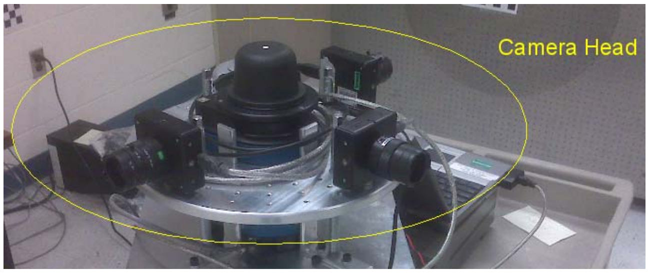
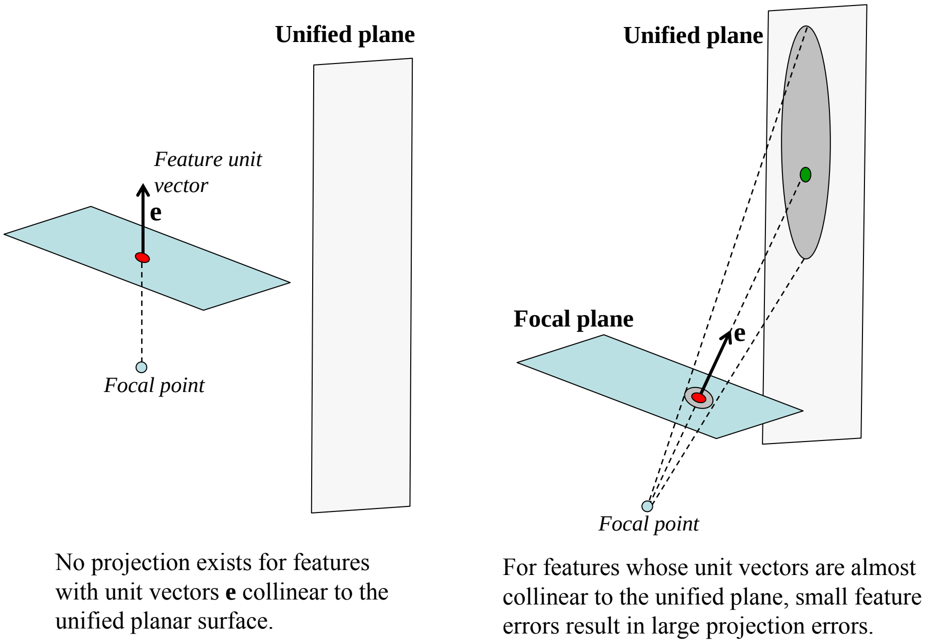


Figure 8: Multi-aperture experimental setup developed by the Alt-Nav team of the Air Force Research Laboratory's Munitions Directorate; the setup includes three video cameras with a 90-deg separation of their optical axes.

A unified coordinate frame has to be chosen for multi-aperture image processing. Particularly, features observed by different cameras must be converted into a single frame of reference. Each camera originally represents its pixels by their two-dimensional (2D) Cartesian coordinates resolved in the camera focal plane (these coordinates are generally referred to as homogeneous coordinates). In principle, a Cartesian representation can be also adapted for the unified frame, which, in this case, will be represented by a planar surface. However, this representation has a drawback of singularity cases as shown in Figure 9.



No projection exists for features with unit vectors **e** collinear to the unified planar surface.

For features whose unit vectors are almost collinear to the unified plane, small feature errors result in large projection errors.

Figure 9: Drawbacks of the Cartesian representation of the unified coordinate frame for the multi-aperture camera case.

Homogeneous feature coordinates extracted from an image of a particular camera must be projected onto a planar surface that is chosen for the multi-aperture frame. For those features whose unit vector (i.e., a unit vector pointed from the camera focal point to the feature) is collinear to the unified plane, such a projection does not exist. If the unit vector is almost collinear to the unified plane, then small errors in feature measurements result in large projection errors.

To remove singular cases associated with the Cartesian representation, a unit sphere approach is chosen for multi-aperture cameras. This approach is motivated by nature’s multi-aperture vision systems, such as the compound eyes of insects, which process optical flow by projecting it onto a sphere. The unit sphere provides a natural framework for wide field-of-view (>180 degree) sensors, where the traditional focal plane representation is not possible. Figure 10 shows the unit sphere feature projection.

Features that are extracted from multiple cameras are projected on the surface of a unit sphere and represented by their spherical azimuth and elevation angles. In Figure 10, the unit sphere origin is collocated with the camera’s focal point. However, this is not required, and the unit sphere projection can be generalized for cases where cameras’ focal points are not collocated with the sphere origin and with each other. Figure 11 illustrates the feature projection for the non-collocated case.

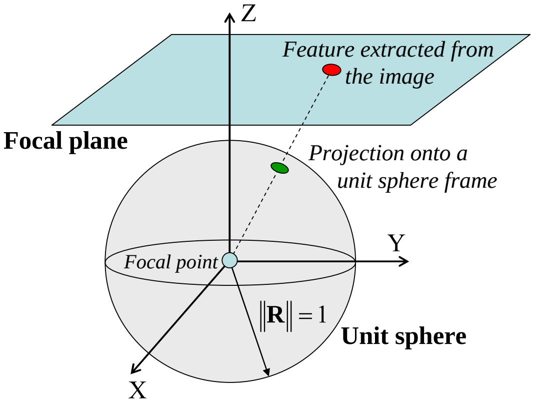

Z
Feature extracted from the image
Focal plane
Projection onto a unit sphere frame
Y
Focal point
$\|\mathbf{R}\| = 1$
Unit sphere
X

Figure 10: Unit sphere frame and unit sphere feature projection.

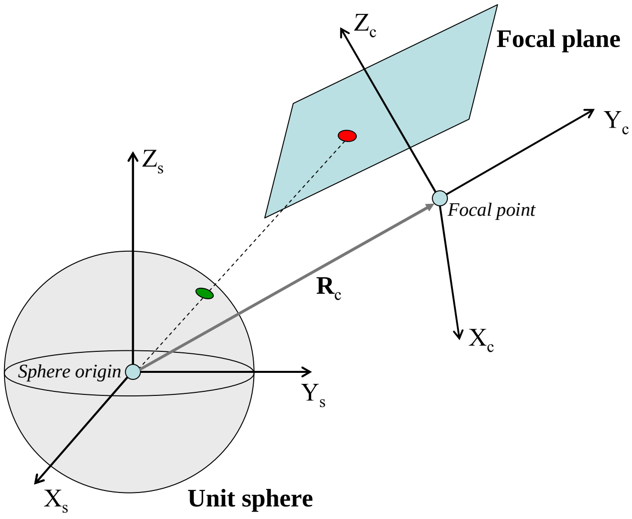


Figure 11: Unit sphere feature projection for the case of non-collocated cameras.

Equation (7) defines unit sphere projections:

$$
\begin{aligned}
&\mathbf{R}_f = C_c^{us} \cdot \left[\tilde{m}_x \quad \tilde{m}_y \quad F\right]^T - \mathbf{R}_c \\
&\mathbf{e}_f = \mathbf{R}_f / \left\|\mathbf{R}_f\right\| \\
&\varphi = \arctan(e_{fy}, e_{fx}) \ \varphi = \arcsin(e_{fz})
\end{aligned}
\tag{7}
$$

where $\mathbf{C}_c^{us}$ is the coordinate transformation matrix from the camera $\{X_c,Y_c,Z_c\}$-frame into the $\{X_s,Y_s,Z_s\}$-frame attached to the unit sphere; $\tilde{m}_x, \tilde{m}_y$ are homogeneous feature coordinates extracted from the image, F is the camera focal length, $\mathbf{R}_c$ is the vector that originates from the unit sphere origin and ends at the camera focal point, and arctan(,) is the four-quadrant arctangent function. The main advantage of the unit sphere formulation is the removal of singularity projection cases that are present if Cartesian representation is utilized (see Figure 9 above).

The use of the unit sphere feature projections requires the transformation of motion equations (i.e., relationships between feature parameters and motion parameters) from the Cartesian frame into the spherical frame. This transformation is illustrated below for a 2D motion case where only a translation motion component is present. In this case, the unified frame is represented by a unit circle as shown in Figure 12.

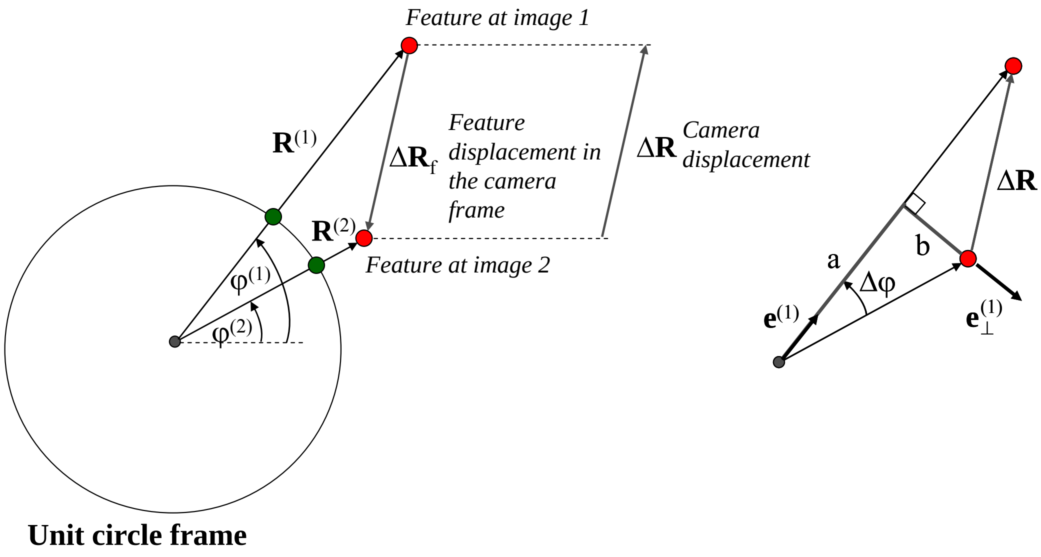


Figure 12: Unit circle representation for a 2D, translation only motion case.

From the geometry shown in Figure 12: $a/\cos\Delta\varphi = b/\sin\Delta\varphi$; where $a = \rho^{(1)} - \left(\Delta\mathbf{R},\mathbf{e}^{(1)}\right)$; $b = -\left(\Delta\mathbf{R},\mathbf{e}_\perp^{(1)}\right)$; $\Delta\varphi = \varphi^{(1)} - \varphi^{(2)}$; (,) is the vector dot product; $\rho^{(1)} = \left\|\mathbf{R}^{(1)}\right\|$ is the distance to the feature for image 1; $\mathbf{e}^{(1)}$ is the feature unit vector for image 1; and, $\mathbf{e}_\perp^{(1)}$ is the unit vector perpendicular to $\mathbf{e}^{(1)}$. Correspondingly, the following constraint equation is formulated:

$$\left(\rho^{(1)} - \left(\Delta\mathbf{R},\mathbf{e}^{(1)}\right)\right)\cdot\cos\Delta\varphi = -\left(\Delta\mathbf{R},\mathbf{e}_\perp^{(1)}\right)\cdot\cos\Delta\varphi \tag{8}$$

This equation can be rearranged as follows:

$$\rho^{(1)}\cdot\cos\Delta\varphi = -\left(\Delta\mathbf{R},\mathbf{e}_\perp^{(2)}\right) \tag{9}$$

Note that $\Delta\varphi$ is directly calculated from camera measurements. Also, equation (9) only includes the unknown range to the feature at the first image. The knowledge of range information for other images (second, third, etc.) is not required. Once the initial range is estimated, the motion constraint (9) can be applied to formulate Kalman filter vision/inertial measurement observables for other images without the need to know (or measure) their feature ranges. As compared to a complete motion formulation that includes feature range variables for the current image (see equation (2) above), the motion constraint formulation is beneficial for a monocular camera case, as well as for stereo cases with a limited baseline, where direct measurements of the range value are not possible or inefficient.

For a general 3D case that includes translational and rotational motion components, two motion constraint equations are derived in a manner similar to the 2D consideration above. The first constraint is obtained by considering the horizontal component of the translational motion, and the second constraint is formulated by adding the vertical motion to it. Constraint equations are then modified to incorporate the rotational motion component. The final constraint equations are as follows:

$$\begin{gathered}\left(\mathbf{e}^{(2)}\right)^{\mathrm{T}}\cdot\Delta\mathbf{C}_{\mathrm{N}}^{\mathrm{b}}\cdot\mathbf{B}\cdot\Delta\mathbf{R}=\left(\mathbf{e}^{(1)}\right)^{\mathrm{T}}\cdot\mathbf{B}^{\mathbf{T}}\cdot\Delta\mathbf{C}_{\mathrm{b}}^{\mathrm{N}}\cdot\mathbf{e}^{(2)}\cdot\rho^{(1)}\\ \left(\mathbf{e}^{(2)}\right)^{\mathrm{T}}\cdot\Delta\mathbf{C}_{\mathrm{N}}^{\mathrm{b}}\cdot\mathbf{D}_{\mathrm{p}}\cdot\Delta\mathbf{R}=\left(\mathbf{e}^{(2)}\right)^{\mathrm{T}}\cdot\Delta\mathbf{C}_{\mathrm{N}}^{\mathrm{b}}\cdot\mathbf{e}_{\perp}{}^{(1)}\cdot\rho^{(1)}\end{gathered}\tag{10}$$

where $\Delta\mathbf{R}$ and $\Delta\mathbf{C}_{\mathrm{b}}^{\mathrm{N}}$ are position and orientation changes between images 1 and 2; and, matrices B and D are defined as:

$$\mathbf{B}=\begin{bmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix},\ \mathbf{D}=\begin{bmatrix}0 & 0 & -\cos(\varphi^{(1)})\\ 0 & 0 & -\sin(\varphi^{(1)})\\ \cos(\varphi^{(1)}) & \sin(\varphi^{(1)}) & 0\end{bmatrix}\tag{11}$$

Motion constraint equations (10) are applied to formulate Kalman filter observables $\eta_{\mathrm{p}}^{\mathrm{Kalman}}$ for the multi-aperture vision inertial aiding:

$$\eta_{\mathrm{p}}^{\mathrm{Kalman}}=\begin{bmatrix}\left(\tilde{\mathbf{e}}_{\mathrm{p}}^{(2)}\right)^{\mathrm{T}}\cdot\Delta\tilde{\mathbf{C}}_{\mathrm{N}}^{\mathrm{b}}\cdot\mathbf{B}\cdot\Delta\tilde{\mathbf{R}}_{\mathrm{INS}}-\left(\tilde{\mathbf{e}}_{\mathrm{p}}^{(1)}\right)^{\mathrm{T}}\cdot\mathbf{B}^{\mathbf{T}}\cdot\Delta\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{N}}\cdot\tilde{\mathbf{e}}_{\mathrm{p}}^{(2)}\cdot\hat{\rho}_{\mathrm{p}}^{(1)}\\ \left(\tilde{\mathbf{e}}_{\mathrm{p}}^{(2)}\right)^{\mathrm{T}}\cdot\Delta\tilde{\mathbf{C}}_{\mathrm{N}}^{\mathrm{b}}\cdot\mathbf{D}_{\mathrm{p}}\cdot\Delta\tilde{\mathbf{R}}_{\mathrm{INS}}-\left(\tilde{\mathbf{e}}_{\mathrm{p}}^{(2)}\right)^{\mathrm{T}}\cdot\Delta\tilde{\mathbf{C}}_{\mathrm{N}}^{\mathrm{b}}\cdot\tilde{\mathbf{e}}_{\perp\mathrm{p}}{}^{(1)}\cdot\hat{\rho}_{\mathrm{p}}^{(1)}\end{bmatrix},\ \mathrm{p}=1,\ldots,\mathrm{P}\tag{12}$$

where $\tilde{\mathbf{e}}_{\mathrm{p}}^{(\mathrm{s})}$, $\mathrm{p}=1,..,\mathrm{P},\ \mathrm{s}=1,2$ is the unit vector of the feature p that is extracted from image s; $\hat{\rho}_{\mathrm{p}}^{(1)}$ is the estimated feature range for image 1, and, $\Delta\tilde{\mathbf{R}}_{\mathrm{INS}}$ and $\Delta\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{N}}$ are INS position and orientation changes between images. Image 1 is generally the image where the feature was first observed and image 2 is the current image.

The initial feature range $\rho^{(1)}$ is estimated by observing the feature from two different locations of the platform and using inertial displacement and orientation change measurements. This estimation is derived from Equation (12) and is formulated as follows:

$$\begin{gathered}\hat{\rho}_{\mathrm{p}}^{(1)}=\left(\mathbf{H}_{\rho}^{\mathrm{T}}\cdot\mathbf{H}_{\rho}\right)^{-1}\cdot\mathbf{H}_{\rho}^{\mathrm{T}}\cdot\begin{bmatrix}\left(\tilde{\mathbf{e}}_{\mathrm{p}}^{(2)}\right)^{\mathrm{T}}\cdot\tilde{\mathbf{C}}_{\mathrm{N}}^{\mathrm{b}}\cdot\mathbf{B}\cdot\Delta\tilde{\mathbf{R}}_{\mathrm{INS}}\\ \left(\tilde{\mathbf{e}}_{\mathrm{p}}^{(2)}\right)^{\mathrm{T}}\cdot\tilde{\mathbf{C}}_{\mathrm{N}}^{\mathrm{b}}\cdot\mathbf{D}_{\mathrm{p}}\cdot\Delta\tilde{\mathbf{R}}_{\mathrm{INS}}\end{bmatrix}\\ \mathbf{H}_{\rho}=\begin{bmatrix}\left(\tilde{\mathbf{e}}_{\mathrm{p}}^{(1)}\right)^{\mathrm{T}}\cdot\mathbf{B}^{\mathbf{T}}\cdot\Delta\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{N}}\cdot\tilde{\mathbf{e}}_{\mathrm{p}}^{(2)}\\ \left(\tilde{\mathbf{e}}_{\mathrm{p}}^{(2)}\right)^{\mathrm{T}}\cdot\Delta\tilde{\mathbf{C}}_{\mathrm{N}}^{\mathrm{b}}\cdot\tilde{\mathbf{e}}_{\perp\mathrm{p}}{}^{(1)}\end{bmatrix}\end{gathered}\tag{13}$$

Essentially, the image depth is initialized using synthetic stereo-vision: camera motion is applied to synthesize a stereo-vision baseline, which is measured by the INS. Note that in this case a correlation between depth estimation errors and inertial position and orientation errors is introduced. This correlation must be taken into account by the Kalman filter design. Particularly, the filter states include INS error states, and errors in initial range estimates. Hence, the correlation between INS position and orientation errors and range estimation errors must be incorporated into the state covariance matrix of the Kalman filter. An approach for computing the INS/vision correlation for the multi-aperture case is similar to the computation of INS/vision correlation for a single aperture case, which is described in [4].

Performance of multi-aperture/vision inertial aiding was assessed in a simulated indoor environment. Figure 13 shows a 2D projection of the simulation environment. The simulation scenario included motion in an indoor hallway. Hallway walls were simulated as vertical with a 2.5-m height. For each wall, four point features were uniformly distributed over its length and height. The simulated motion trajectory was two-dimensional. However, the vision/inertial system estimated all six degrees of motion freedom. Inertial measurement unit was simulated as a lower-grade unit with an accelerometer bias of 1 mg and a gyro drift stability of 50 deg/hr. Specifications of video cameras were simulated as follows: 640x480 resolution, 10 Hz update rate, 40 deg by 30 deg field-of-view (FoV). Errors of extracting a point feature from camera images were simulated as a camera quantization error plus a random Gaussian noise error with a standard deviation of one pixel.

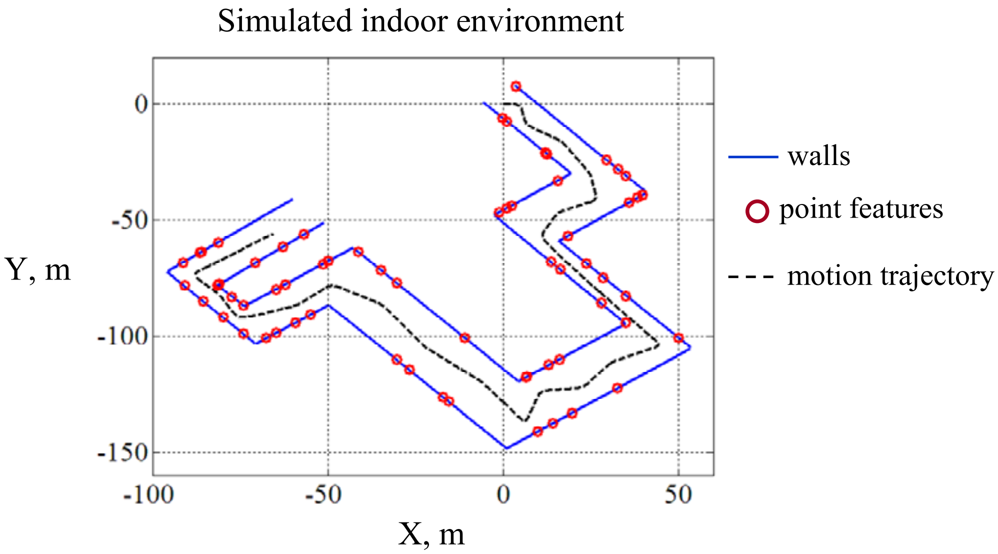
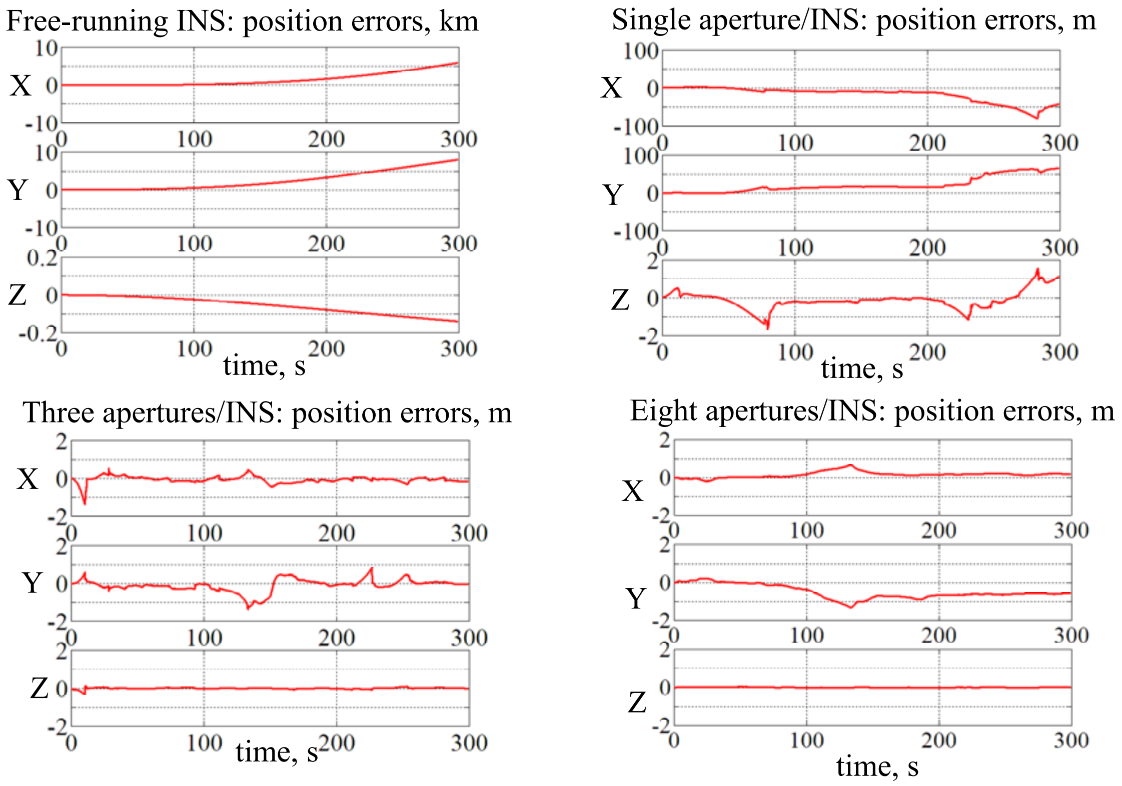


Figure 13: Simulated indoor environment implemented for the evaluation of multi-aperture vision/INS integration.

Figure 14 exemplifies simulation results. Position accuracy performance is shown for cases of a free-running inertial, a single camera/INS integration, and a multi-aperture vision/INS integration including cases of three and eight apertures. For the multi-aperture cases, the angular separation between individual cameras was implemented as 120 deg and 45 deg for the three-aperture and eight-aperture cases, accordingly.


Free-running INS: position errors, km

Single aperture/INS: position errors, m

Three apertures/INS: position errors, m

Eight apertures/INS: position errors, m

Figure 14: Example simulation results for the indoor simulation environment.

Simulation results presented demonstrate that for the simulation scenario implemented the use of multi-aperture INS aiding allows reducing position errors by two orders of magnitude as compared to the single-aperture case: position drift is reduced from a 100-m level to a 1-m level. The accuracy performance is improved due to the following factors. Firstly, the number of available features generally increases by increasing the camera FoV using multiple apertures. Secondly, and more importantly, the use of multiple apertures enhances the feature range initialization capabilities. Figure 15 illustrates this effect. The range estimation accuracy is primarily influenced by the feature angular change between images that are used for the estimation. The determinant of the $\mathbf{H}_\rho^T \cdot \mathbf{H}_\rho$ matrix in the pseudo-inverse formulation of equation (13) is directly related to changes in spherical angles. An increased angular change increases the determinant value and thus improves the estimation accuracy. The maximum angular change is accumulated when passing over a feature and observing this feature with a side-looking camera as shown in Figure 15. As a result, this feature is efficiently initialized after a small platform displacement. Kalman filter observables for this feature are then applied to correct inertial drift over longer displacement intervals that are required in order to accumulate an angular change that is sufficient to initialize the feature range for the front-looking camera.

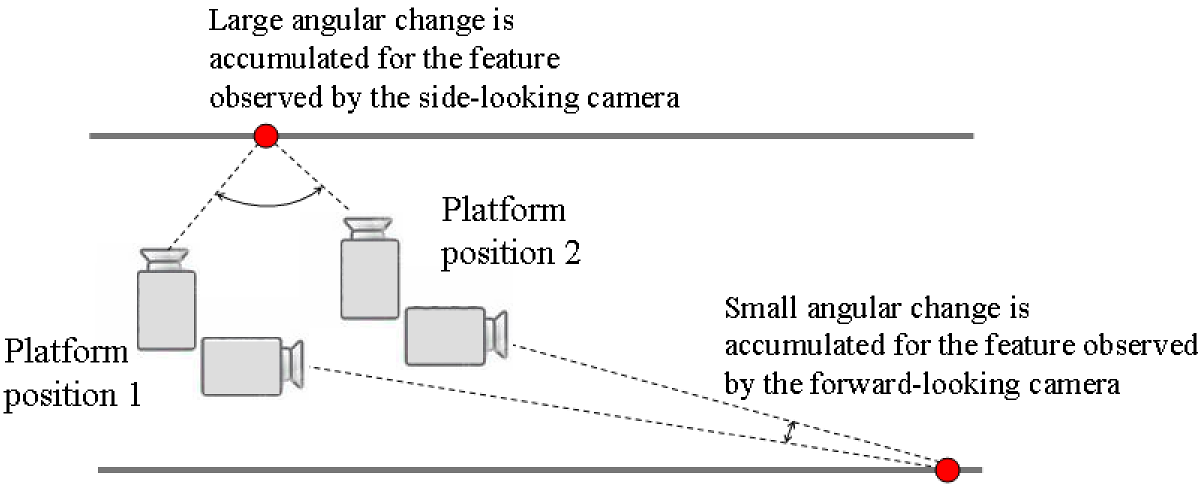

Large angular change is accumulated for the feature observed by the side-looking camera
Platform position 2
Small angular change is accumulated for the feature observed by the forward-looking camera
Platform position 1

Figure 15: Improved range initialization capabilities using multi-aperture vision: large angular change is accumulated over a small distance for the feature observed by the side-looking camera; this feature is initialized and then applied to correct inertial drift over a longer distance that is required to accumulate an angular change sufficient for initializing the feature observed by the front-looking camera.

3.0 LADAR-AIDED INS

Whereas image-aided navigation uses passive electro-optical sensors, laser-based navigation makes use of active sensors: laser scanners (Ladars) or imaging lasers. The use of Ladar for navigation purposes has been demonstrated for both en-route navigation [22] and precision approach guidance [23] in a laser-based TRN and has been shown to provide meter-level accuracies [2]. However, TRN techniques are operationally limited by the availability of an on-board terrain database at the location of interest. More recently, a method has been proposed to perform the navigation function using two airborne scanning Ladars integrated with an INS [24]. Due to the lack of terrain in urban and indoor environments, the Ladar-based methods must exploit features such as surfaces, corners, points, etc. For the feature-based navigation, changes in position and orientation are estimated from changes in the parameters of features that are extracted from Ladar images. Two-dimensional (2D) laser scanners and feature-based localization methods have been used extensively to enable navigation of robots in an indoor environment. For example, reference [25] describes a method to estimate the translation and rotation of a robot platform from a set of extracted lines and points using a 2D sensor. Reference [26] discusses the feature extraction and localization aspects of mobile robots and addresses the statistical aspects of these methods, whereas reference [27] introduces improved environment-dependent error models and establishes relationships between the position and heading uncertainty and the laser observations, thus enabling a statistical assessment of the quality of the estimates. In [28], 2D scanning LADAR measurements are tightly integrated with IMU measurements to estimate the relative position of a van in an urban environment. The idea of using 3D measurements and planar surfaces for 2D localization is introduced in [29]. Note that the above applications focus on 2D navigation (two position coordinates and a platform heading angle). However, for applications such as autonomous aerial vehicles, a 3D navigation solution is required, especially, for those cases where the platform attitude varies in pitch, roll, and yaw directions. To enable 3D navigation, the utilization of the laser range scanner measurements must, somehow, be expanded for estimation of 3D position and attitude. The use of 3D features from 3D imaging cameras was introduced in [30]. The discussion in this paper will limit itself to 2D and 3D navigation using 2D scanning Ladar sensors.

3.1 Basic Concept

Figure 16 illustrates the basic measurement mechanism that underlies laser scanner-based navigation. The system uses a scanning Ladar to make range and angular measurements of its environment and converts these "raw" measurements to a set of points in a two-dimensional (2D) Ladar coordinate frame using a polar coordinate transformation. Features are extracted from Ladar scans at different times and used to determine a change in position and orientation of the user. Data from scanning Ladar sensors can support both 2D [28] and, by enabling gimballing of the sensor, three-dimensional (3D) navigation [31].

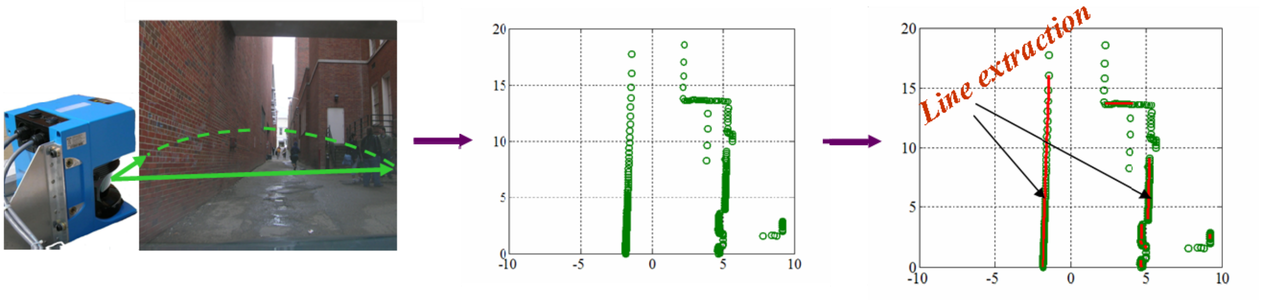


Figure 16: Laser Scanner-Based Navigation.

The discussion in this paper will focus on 3D navigation since 2D navigation (with line segments) is a considered a special case of 3D navigation. The features that are exploited are "lines" for 2D navigation and "planar surfaces" for 3D navigation. The rationale for the use of lines and planar surfaces for navigation in 3D urban environments is that these features are common in man-made environments. To exemplify this, Figure 17 shows images of typical urban indoor (hallway) and outdoor (urban canyon) environments. Multiple planes can be observed in both images. Since changes in image feature parameters between scans at different time epochs are used for navigation, the feature must be observed in both scans. Feature repeatability is thus essential for Ladar-based navigation. Planar surfaces satisfy this requirement, as they are highly repeatable from scan to scan. As long as a wall of a building stays within the Ladar measurement range and FoV, the plane (or line) association process is likely to be successful.

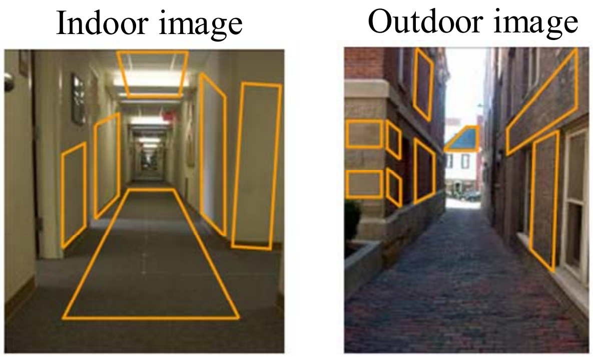


Figure 17: Examples of planar surfaces observed in urban images: multiple planes can be extracted for indoor and outdoor image examples.

Figure 18 shows a block diagram of the Ladar-based navigation method. The desired features ("lines" or "planar surfaces") are extracted from the Ladar scans and used to estimate the navigation solution that is comprised of changes in Ladar position and orientation. In order to use a planar surface for the estimation of position and orientation changes, this planar surface must be observed in both scans and it must be known with certainty that the plane in one scan corresponds to the plane in the next scan. Hence, the feature association (or matching) procedure establishes a correspondence between planes extracted at the current scan time epoch and the planes extracted at previous scan epochs. The navigation routine stores planes extracted from previous scans into the plane list, which is initialized with the planar surfaces of the first scan. If a new planar surface is observed during one of the following scans, the plane list is updated to include this new plane. Reference [28] discuses how INS data is used to match extracted "line" features for the 2D navigation case. In order to use INS data for plane matching, the line-matching algorithms developed in [28] were extended to the 3D case. The feature matching procedure uses position and orientation outputs of the INS to predict the planar surface location and orientation at the current scan time based on planar surface parameters observed at previous scan time epochs. If the predicted planar surface parameters and the parameters of one of the extracted planar surfaces match closely, a successful association is declared. Following successful feature association changes in the associated planar surface parameters are used to estimate the navigation and attitude solution. Furthermore, these changes in planar surface parameters are also applied to periodically re-calibrate the INS to reduce drift terms in inertial navigation outputs in order to improve the quality of the INS-based plane prediction used by the feature matching procedure.

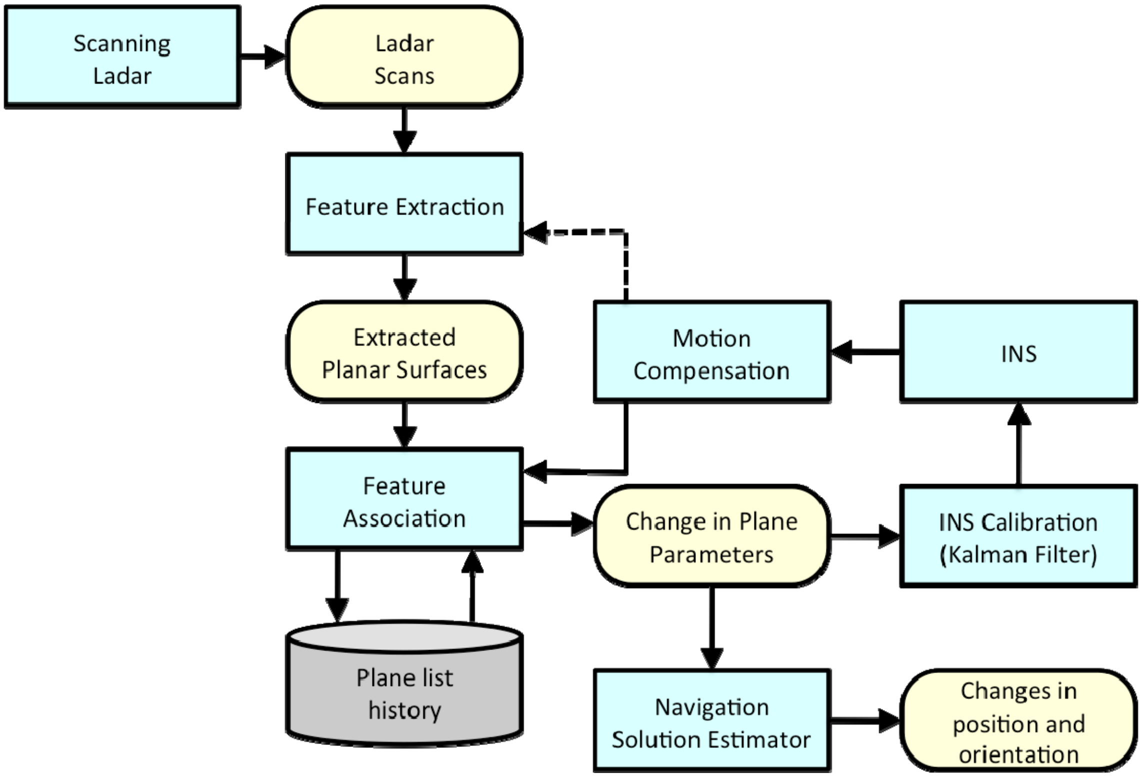

Scanning Ladar
Ladar Scans
Feature Extraction
Extracted Planar Surfaces
Motion Compensation
INS
Feature Association
Change in Plane Parameters
INS Calibration (Kalman Filter)
Plane list history
Navigation Solution Estimator
Changes in position and orientation

Figure 18: Generic routine of 3D navigation that uses images of a scanning LADAR.

The position and orientation data from the INS is also used to compensate for the motion of the platform (Ladar) during the scans in those cases where such motion could introduce significant distortions to Ladar scan images. The above method results in the calculation of a relative navigation solution. Estimating local frame position and orientation in one of commonly used navigation frames (e.g. East-North-Up and Earth-Centered-Earth-Fixed frames) allows for the transformation of the relative navigation solution into an absolute navigation solution. Navigation herein is performed in completely unknown environments. No map information is assumed to be available *a priori*.

3.2 Mechanization

Planar surface based navigation is performed using the perceived location changes of the normal point on the planar surface between scans at different time epochs. A planar surface normal point is defined as the intersection of the plane and a line originating from the Ladar location perpendicular to the plane of interest. Figure 19 illustrates the geometry involved in navigating off the plane normal points.

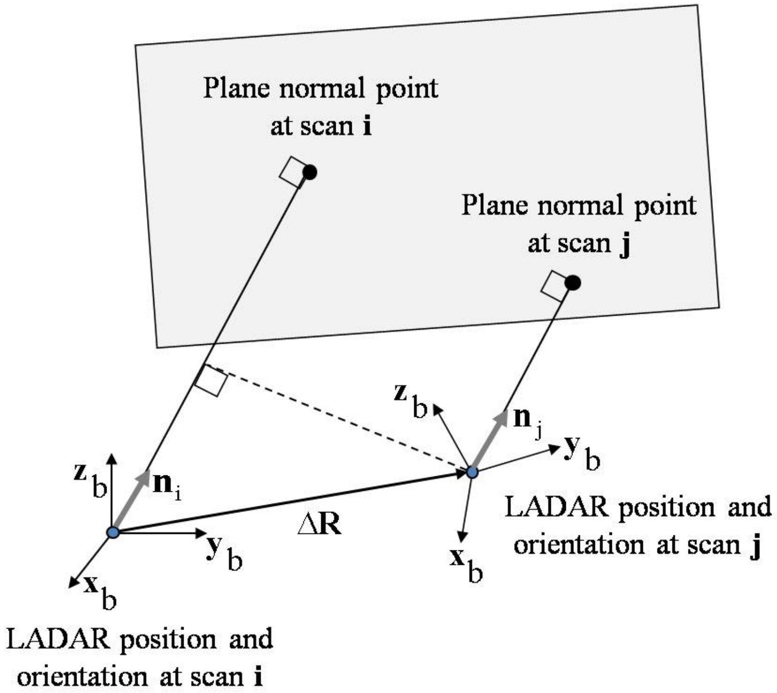


Figure 19: Use of plane normal points for navigation: changes in perceived location of the normal point between scan i and scan j are applied to estimate position changes.

In Figure 19, Δ**R** is the delta position vector (displacement vector between scans i, at time t_i, and scan j, at time t_j, in this case); $\mathbf{n}_i$ is the plane normal vector whose components are resolved in the Ladar body frame at scan epoch t_i; $\mathbf{n}_j$ is the plane normal vector whose components are resolved in the Ladar body frame at scan epoch t_j ; and, ρ_i and ρ_j are the shortest distances from the Ladar to the plane at epochs t_i and t_j , respectively. These distances are referred to as the normal point distances. Note that in the navigation frame, the planar surface normal vectors at epochs t_i and t_j are equal since stationary planar surfaces are assumed. However, expressed in the Ladar body frame both normal vectors are likely to be unequal due to the body frame rotation between epochs t_i and t_j. From the geometry presented in Figure 19, a relationship can be derived between the projection of the displacement vector (between epochs t_i and t_j) onto the planar surface normal vector and the change in the normal point range between scans i and j:

$$\Delta\mathbf{R}\cdot\mathbf{n}_i = \rho_i - \rho_j \tag{14}$$

Given M associated planar surfaces, a set of linear equations like (14) can be set up in matrix form:

$$\mathbf{H}\cdot\Delta\mathbf{R} = \Delta\rho \tag{15}$$

where:

$$\mathbf{H} = \begin{bmatrix} \mathbf{n}_{i,1}^T \\ \vdots \\ \mathbf{n}_{i,K}^T \end{bmatrix}, \quad \Delta\rho = \begin{bmatrix} \rho_{i,1} - \rho_{j,1} \\ \vdots \\ \rho_{i,K} - \rho_{j,K} \end{bmatrix} \tag{16}$$

Note that a minimum of three non-collinear planar surfaces is required for the observation matrix, **H,** to be non-singular and thus allowing for a unique solution of equation (16). Using Ordinary Least Squares (OLS), a solution can be computed from equation (16) as follows:

$$\Delta\hat{\mathbf{R}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\Delta\hat{\rho} \tag{17}$$

In equation (17), $\Delta\hat{\rho}$ is the estimated delta range vector obtained from the planar surface extraction and parameter estimation process. For 2D navigation, “lines” are used and assumed to be part of a vertical planar surface where possible tilt of the vehicle or planar surface is estimated separately [28]. In 3D navigation the Ladar scanner is rotated intentionally and scans at three (or more) discrete and known elevations are used to estimate the planar surface parameters [31].

The attitude of the Ladar platform is estimated using the planar surface normal vectors. Equation (18) relates a change in the Ladar orientation from time epoch t_j to t_i , given by DCM $\mathbf{C}_j^i$, to the change in the orientation of the normal vectors:

$$\mathbf{C}_j^i\mathbf{n}_j = \mathbf{n}_i \tag{18}$$

To compute the DCM, the attitude estimation algorithm needs to solve Wahba’s problem [32]: given a first set of normal vectors with vector components resolved in the Ladar frame at t_i and the second set of the same vectors with their components resolved in the Ladar frame at t_j, find the DCM that brings the second set into best least square correspondence with the first. At least two non-collinear normal vectors are required for the attitude estimation. Attitude is generally estimated by solving an eigenvalue/eigenvector problem, which requires solution of non-linear equations. Solution approaches have been described by various sources such as [33], [34] and [35].

In order to use planar surfaces for the estimation of position and orientation changes as formulated earlier in this section, it must be known with certainty that a plane at time epoch t_i corresponds to the plane in at time epoch t_j. To enable association of planar features, INS data can be used to predict the planar surface range and normal vector at time t_j based on range and normal vector estimated at time t_i:

$$\begin{aligned} \hat{\rho}_j^- &= \hat{\rho}_i - \left(\Delta\tilde{\mathbf{R}}_{INS}, \mathbf{n}_i\right) \\ \hat{\mathbf{n}}_j^- &= \Delta\tilde{\mathbf{C}}_i^j \cdot \hat{\mathbf{n}}_i \end{aligned} \tag{19}$$

where $\hat{\rho}_j^-$ and $\hat{\mathbf{n}}_j^-$ are the predicted range and normal vector; $\hat{\rho}_i$ and $\hat{\mathbf{n}}_i$ are plane range and normal vector extracted at time i; and, $\Delta\tilde{\mathbf{R}}_{INS}$ and $\Delta\tilde{\mathbf{C}}_i^j$ are the INS measurements of the position change vector and DCM increment from time epoch t_i to time epoch t_j . If the predicted range and normal vector ($\hat{\rho}_j^-$ and $\hat{\mathbf{n}}_j^-$) match closely to the range and normal vector extracted from scan j ($\hat{\rho}_j$ and $\hat{\mathbf{n}}_j$), the planar surface correspondence is established between scans i and j. Note that planar surface matching thresholds must accommodate both planar surface extraction errors and INS drift errors.

Ladar data is also used for the in-motion calibration of the inertial error states in order to reduce drift of the INS navigation outputs. The INS calibration part and Ladar-based position computations are separated. Since the INS is calibrated in the Ladar measurement domain, the integration mechanization is considered tightly coupled. Similar to the case of GPS/INS integration, one of the main advantages of tight coupling over a loosely coupled position-domain INS calibration is the ability to reduce inertial drift for those cases where Ladar scans do not contain enough features to compute a laser-based position. In those cases where not enough features are present for a Ladar-based position solution, the navigator of Figure 18 will employ inertial coasting.

The dynamic-state INS calibration uses a Kalman filter to periodically estimate inertial error states. The estimation process is based on a complementary Kalman filter methodology [21], which employs differences between INS and laser scanner observables as filter measurements. Changes in planar surface ranges between consecutive scans are used as laser observables. Correspondingly, laser scanner observables of the Kalman filter are formulated as follows for the scan at time epoch tm:

$$\Delta\rho_{LS}(t_m) = \begin{bmatrix} \rho_1(t_{m-1}) - \rho_1(t_m) \\ \ldots \\ \rho_K(t_{m-1}) - \rho_K(t_m) \end{bmatrix} \tag{20}$$

where K is the number of features for which the match is found between time epochs tm and tm-1.

Equivalent observables can be synthesized from INS measurements by transformation of the INS displacement vector into the range domain as follows:

$$\Delta\rho_{INS}(t_m) = \mathbf{H}(t_{m-1})\left(\Delta\tilde{\mathbf{R}}_{INS}(t_m) + \left(\tilde{\mathbf{C}}_b^N(t_m) - \tilde{\mathbf{C}}_b^N(t_{m-1})\right)\mathbf{l}_b\right) \tag{21}$$

where $\mathbf{H}(t_{m-1})$ is the observation matrix at time epoch m-1 given by equation (9), $\Delta\tilde{\mathbf{R}}_{INS}(t_m)$ is displacement from time epoch m-1 to m calculated from the IMU measurements, and $\mathbf{l}_b$ is the lever arm vector pointed from the IMU to the Ladar scanner with the vector components given at the laser scanner-defined body frame. As mentioned previously, filter measurements are defined as differences between inertial and laser scanner observables:

$$\mathbf{y}_{Kalman}(t_m) = \Delta\rho_{INS}(t_m) - \Delta\rho_{LS}(t_m) \tag{22}$$

The filter operates with dynamic states only (position error states are not used). Particular filter states include: errors in position changes between consecutive scans, velocity errors, attitude errors, gyro biases, and accelerometer biases:

$$\delta \mathbf{x} = \left[\delta\Delta\mathbf{R}_N^T \quad \delta\mathbf{v}_N^T \quad \boldsymbol{\psi}^T \quad \mathbf{a}_b^T \quad \mathbf{b}_b^T \right]^T \tag{23}$$

For this state vector, the observation matrix $\mathbf{H}_{\text{Kalman}}$ can be derived directly by augmenting the geometry matrix of equation (16) with zero elements:

$$\mathbf{H}_{\text{Kalman}}(t_m) = \begin{bmatrix} \mathbf{n}_1^T(t_{m-1}) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{n}_K^T(t_{m-1}) & 0 & \cdots & 0 \end{bmatrix} \tag{24}$$

The measurement noise matrix $\mathbf{R}_{\text{Kalman}}$ is derived from the line and planar surface estimation processes performing a comprehensive covariance analysis of the feature extraction method [27][28][31].

Derivation of the filter state transition matrix and the system noise matrix for the filter states chosen employs a standard Kalman filter formulation, which can be found in [21]. Now, the complementary Kalman filter can be implemented to estimate the inertial error states in a manner similar to the once discussed in the image-aided navigation section using the filter states, the filter measurements, and the filter matrices. Note that the Kalman filter implementation considered herein does not take into account a correlation between errors in estimated ranges of extracted line normal points and inertial attitude errors.

It is important to mention that the addition of new lines to the line list generally improves the feature geometry and availability but at the same time creates a drift in the relative position solution. If the same lines are seen in the initial scan and the following scans then errors in the relative position estimate are due to line extraction errors in the current and initial scans only. For this case, the position solution does not drift. If a new line is seen, it is transformed into the initial frame using the current position estimate. Errors in the position estimate are thus transformed into errors in line parameters and add up to line extraction errors. As a result, the current position error contributes to the position error for the next scan where the new line is used for navigation. A position drift is thus created. This position drift is however generally significantly smaller as compared to the case where position increments are estimated based on consecutive scans thus introducing a random walk.

Parameters of a planar surface can be estimated by exploiting motion of a 2D Ladar scanner in order to observe the surface from different points of view. This is illustrated in Figure 20. Now the multiple observed lines can be transformed to a common frame using the INS data and associated to form lines belonging to the same planar surface. Multiple of these lines can then be used to estimate the planar surface parameters. More details of this method can be found in [31].

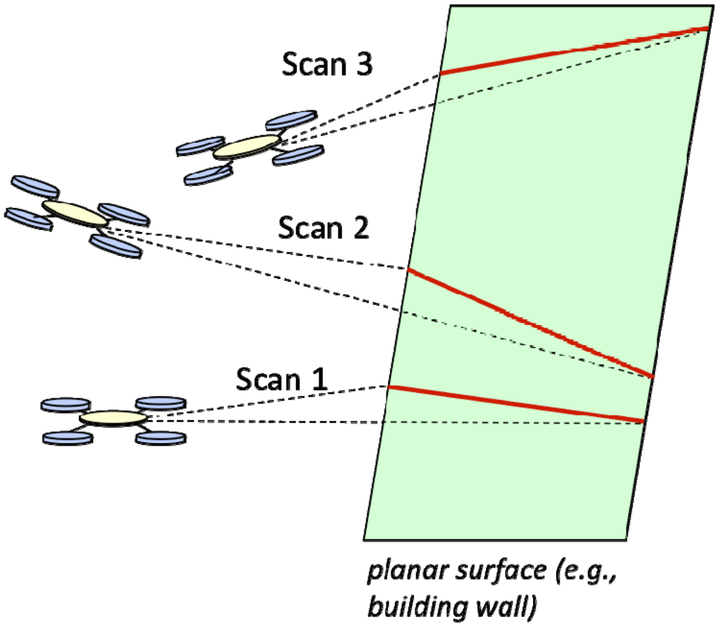


Figure 20: Estimating planar surface parameters using 2D Ladar scans and the Ladar's motion.

3.3 Performance Demonstration

To show the performance of the algorithms discussed in Section 3.2 for 2D and 3D navigation, three sets of results are shown. In all three cases a SICK LMS-200 Ladar scanner and a Systron Donner DQI tactical grade IMU were used. For the first test the equipment was mounted on a test cart that was manually pushed through hallways of the Ohio University Stocker Engineering Center. To create a reference trajectory, marks were put on the floor along the test path as illustrated by Figure 21. These marks were used as reference points. The cart was stopped at each reference point and the laser/inertial-integrated readings were compared with reference point locations. The true trajectory and the estimated trajectories are shown in the graph in Figure 21. The relative position errors of the indoor test were found to be limited to 10 cm after traveling a distance of 9 m (or 1.1% of distance traveled). A cm-accurate accurate trajectory reconstruction is thus demonstrated for the indoor test scenario considered.

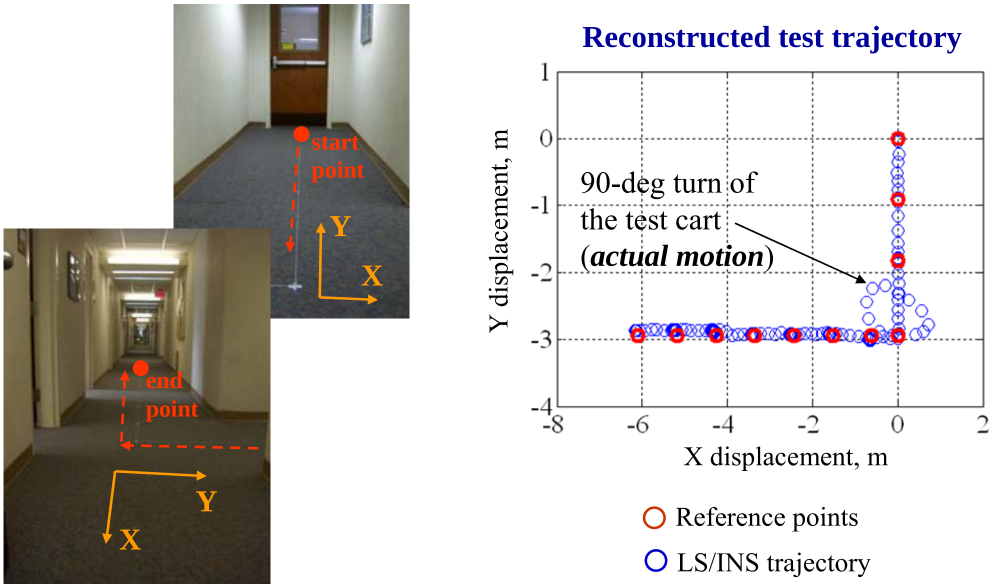


Figure 21: Indoor hallway test environment and test results.

In the second scenario, the test setup was mounted on a van and driven through some of the more challenging urban environment of Athens, Ohio. Since GPS was unavailable on much of the trajectory, a good truth reference could not be established. Hence, the overall performance was measured by investigating the residual errors of the Kalman filter as well as the overall drift after each of the scenarios. This drift error was estimated by computing the van's beginning and end position using kinematic GPS and computing the difference between the end position from the survey and the end-position as calculated by the algorithm. Figure 22 shows the scenario for the urban environment scenario and the trajectory as estimated by the algorithm. The drift error over the whole trajectory was determined to have a magnitude equal to 1.59 m.

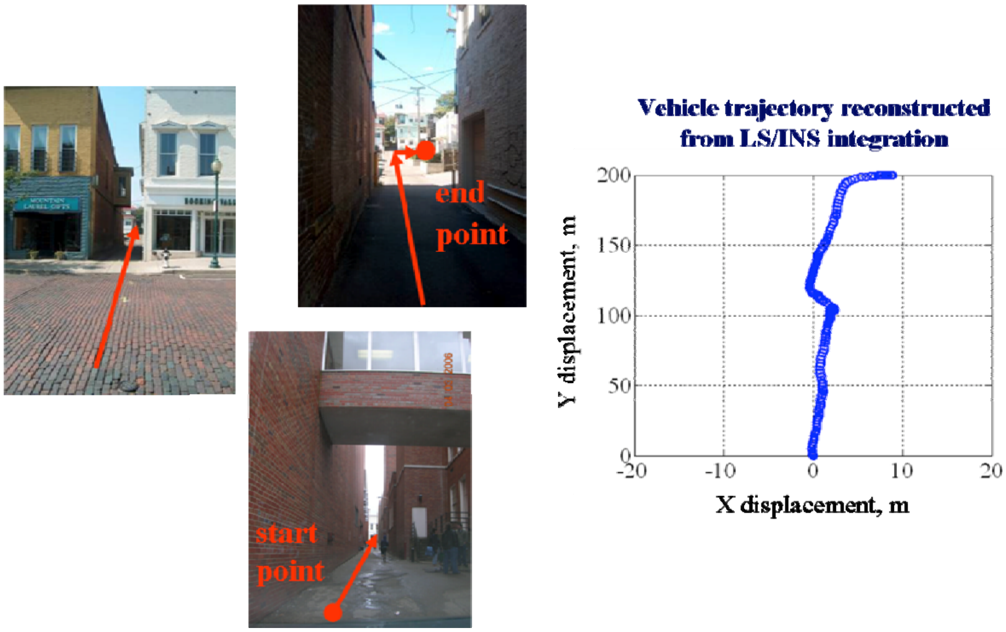


Figure 22: Outdoor urban environment scenario and van trajectory.

Finally, the 3D navigation scenario was evaluated in a laboratory environment using actual test equipment as shown in Figure 23. Position errors shown are at the cm-level. Standard deviations of the errors in x, y, and z position components are computed as 3.5 cm, 0.7 cm, and 2.1 cm, respectively. As shown in more details in [31], orientation measurements were made as well. The standard deviation of the errors in pitch, roll, and heading angles are estimated to be 1.6 deg, 2.5 deg, and 1 deg, respectively. It is noted that attitude errors for the live data test are increased notably as compared to attitude errors that were done using simulations (from sub-degree level to degree level). This error increase is mainly attributed to the deviation between the commanded angular value and the actual elevation angle of the low-cost servomotor used in the test setup.

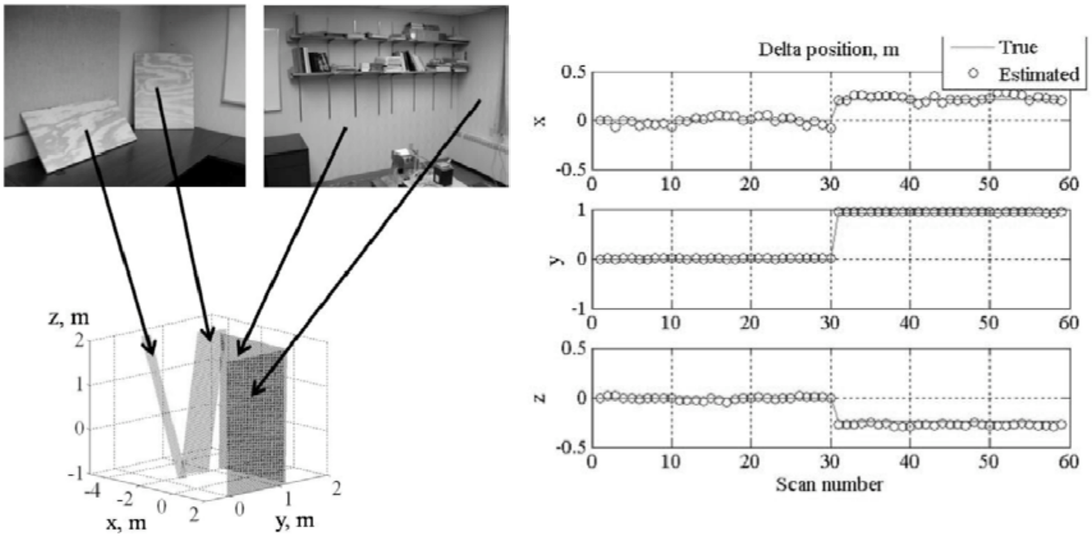


Figure 23: Test environment, planar surfaces and the reconstruction of 3D translational motion: true trajectory vs. motion trajectory estimated by the 3D planar-based navigation algorithm.

4.0 A LOOK INTO THE FUTURE

Advances in navigation sensor and integration technologies have allowed us to equip a wide variety of platforms with an integrated navigation capability that has not been cost effective in the past. The modern commander demands precise location information of operational assets, be they aircraft, ships, spacecraft, ground vehicles (both manned and robotic) as well as individual soldiers. This capability will enhance situational awareness and command and control, increase speed of maneuver, improve weapon effectiveness, and simplify communications and logistics requirements.

Navigation systems will continue to get progressively smaller and more affordable. Technological advances, improved mass production techniques and further cost reductions are forecast for the foreseeable future for both inertial sensors and satellite systems. Integrated navigation systems, such as INS/GPS, will continue to be the standard and applicable to almost any platform or application. These systems will provide accurate position, attitude, velocity and extremely accurate time information to the user continuously. New sensor technologies, innovative methods for exploiting signals of opportunity, bio-inspired navigation concepts, and improved data integration algorithms are all actively being pursued to handle the difficult case of navigating when satellite navigation is not always available. This will be particularly useful for ground forces operating in complex urban, indoor and subterranean environments.

Some of the new and emerging military applications of future integrated navigation systems will include:

- Networking of sensor platforms to create new and highly effective integrated network sensor systems.
- Robotic assets that use precise navigation systems to allow autonomous maneuver, alone or in concert with others.
- Guided munitions (e.g. artillery shells) that can attack a given coordinate precisely, independent of aiming accuracy will soon be a reality.

- Accurate positioning of all platforms even when operating in GPS denied environments.
- Other innovative applications that we have not yet imagined.

5.0 SUMMARY

Navigating in indoor and highly urban locations is a “navigation gap” where GPS cannot currently perform, and the use of Alt-Nav techniques is one potential way to fill that navigation gap. The concept of integrated navigation is to combine the outputs of different types of sensors (see Figure 24). As we have discussed, navigation sensors, like any system, have strengths and weaknesses. For example, GPS has exceptional accuracy, but it is subject to outages due to the loss of satellite signals. Inertial sensors rely only on gravity and platform dynamics, which cannot be jammed, but they exhibit errors that grow over time and eventually become unacceptable. Through the careful design of the integrated system, the limitations of individual sensor technologies can be greatly mitigated and improved accuracy and robustness can be achieved.

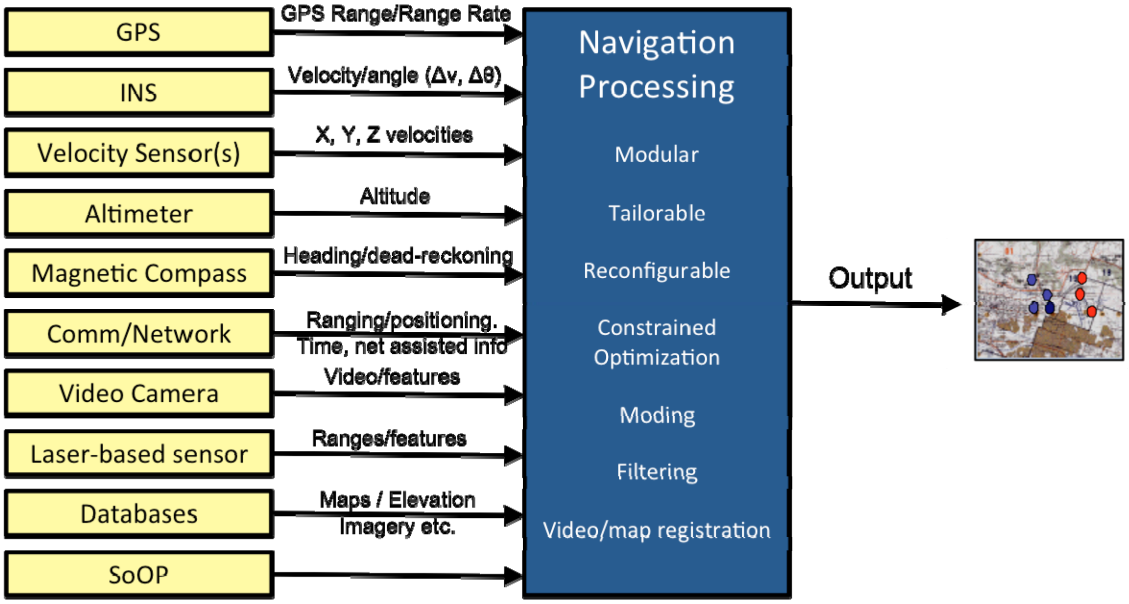


Figure 24: Modular, adaptive, multi-sensor integrated systems provide a possible solution to providing robust navigation to dismounted soldiers in complex terrain.

DISCLAIMER

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S Government.

REFERENCES

[1] Kaser, J. and C. Hutchinson, “The OMEGA Navigation System—An Overview,” I*EEE Communication Society Magazine*, pp. 23-35, May 1978.

[2] Duckworth, G. L. and E. J. Baranoski, "Navigation in GNSS-denied Environments: Signals of Opportunity and Beacons," *Proceedings of the NATO Research and Technology Organization (RTO) Sensors and Technology Panel (SET) Symposium*, Antalya, Turkey, 1-2 October 2007.

[3] Veth, M. and J. Raquet, "Fusion of Low-Cost Imaging and Inertial Sensors for Navigation," *NAVIGATION - Journal of the Institute of NavigationVol. 54, No. 1, Spring 2007*.

[4] Veth, M. J., "Fusion of Imaging and Inertial Sensors for Navigation," Dissertation, Air Force Institute of Technology (AFIT), September 2006.

[5] Campbell, J., M. Miller, M. Uijt de Haag, D. Venable, and M. Smearcheck, "Flash-LADAR Inertial Navigator Aiding," *Proceedings of IEEE/ION PLANS 2006*, San Diego, CA, pp. 677-683, April 2006.

[6] Soehren, W. and W. Hawkinson, "A Prototype Personal Navigation System," *Proceedings of IEEE/ION PLANS 2006*, San Diego, CA, pp. 539-546, April 2006.

[7] Barnes, J., C. Rizos, M. Kanli, D. Small, G. Voigt, N. Gambale, J. Lamance, T. Nunan, C. Reid, "Indoor Industrial Machine Guidance Using Locata: A Pilot Study at BlueScope Steel," *Proceedings of 2004 ION Annual Meeting*, pp. 533-540, June 2004.

[8] Opshaug, G. and P. Enge, "GPS and UWB for Indoor Positioning," *Proceedings of ION GPS-2001*, Salt Lake City, UT, pp. 1427-1433, September 2001.

[9] M. Rabinowitz and J. Spilker, "The Rosum Television Positioning Technology," *Proceedings of 2003 ION Annual Meeting*, Albuquerque, NM, pp. 527-541, June 2003.

[10] Eggert, R. and J. Raquet, "Evaluating the Navigation Potential of the NTSC Analog Television Broadcast Signal," *Proceedings of ION GNSS-2004*, Long Beach, CA, pp. 2436-2446, September 2004.

[11] Hall, T, C. Counselman III, P. Misra, "Radiolocation Using AM Broadcast Signals: Positioning Performance," *Proceedings of ION GPS-2002*, Portland, OR, September 2002.

[12] McEllroy, J., J. Raquet, and M. Temple, "Use of a Software Radio to Evaluate Signals of Opportunity for Navigation," *Proceedings on ION GNSS-2006*, Fort Worth, TX, pp. 126-133, September 2006.

[13] Miller, M. M., et al., "Navigating in Difficult Environments: Alternatives to GPS – 2," *Proceedings of the NATO RTO Lecture Series on "Low Cost Navigation Sensors and Integration Technology,"* SET-116, 2008.

[14] Hrabar, S. and Sukhatme, G. S., A Comparison of Two Camera Configurations for Optic-Flow Based Navigation of a UAV Through Urban Canyons, In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Volume 3, pp. 2673-2680, September 2004.

[15] Adams, H., Singh, S. and Strelow, D. An Empirical Comparison of Methods for Image-based Motion Estimation, In Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Volume 1, pp. 123-128, September 2002.

[16] Olson, C. F., Matthies, L. H., Schoppers, M. and Maimone, M. W., Robust Stereo Ego-motion for Long Distance Navigation, In Proceedings of the IEEE Conference on Advanced Robotics, Volume 2, pp. 453-458, June 2000.

[17] Pachter, M. and Porter, A., Bearings-only Measurements for INS Aiding: The Three-Dimensional Case, In Proceedings of the 2003 AIAA Guidance, Navigation and Control Conference, 2003, AIAA paper number 2003-5354.

[18] Hagen, E. and Heyerdahl, E., Navigation by Optical Flow, In Proceeding of the 11th IAPR International Conference on Pattern Recognition, Volume 1, pp. 700-703, 1992.

[19] Lowe, D. G., Object recognition from local scale-invariant features. In Proc. of the International Conference on Computer Vision, volume 2, pp. 1150-1157, September 1999. Corfu, Greece.

[20] Maybeck, Peter S. *Stochastic Models Estimation and Control, Vol I.* Academic Press, Inc., Orlando, Florida 32887, 1979.

[21] Brown, R. G. and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, Third Edition, John Wiley & Sons, Inc., New York, 1997.

[22] Campbell, J. L., M. Uijt de Haag, and F. van Graas, "Terrain-Referenced Positioning Using Airborne Laser Scanner," *NAVIGATION Journal of the Institute of Navigation*, Vol. 52, No. 4, pp. 189-197, 2005-2006.

[23] Campbell, J. L., M. Uijt de Haag, F. van Graas, "Terrain-Referenced Precision Approach Guidance: Proof-of-Concept Flight Test Results," *NAVIGATION - Journal of the Institute of Navigation*, Volume 54, Number 1, Spring 2007.

[24] Vadlamani, A. K. and M. Uijt de Haag, "Use of Laser Range Scanners for Precise Navigation in Unknown Environments," in Proceedings of the Institute of Navigation GNSS-2006, pp. 1104-1114, September 2006.

[25] Borges, G. A. and M.-J. Aldon, "Optimal Robot Pose Estimation using Geometrical Maps," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 1, pp. 87-94, February 2002.

[26] Pfister, S. T., "Algorithms for Mobile Robot Localization and Mapping Incorporating Detailed Noise Modeling and Multi-scale Feature Extraction," Ph.D. Dissertation, California Institute of Technology, 2006.

[27] Bates, D., "Navigation Using Optical Tracking of Objects at Unknown Locations," M.S. Thesis, Ohio University, 2006.

[28] Soloviev, A., D. Bates, and F. van Graas, "Tight Coupling of Laser Scanner and Inertial Measurements for a Fully Autonomous Relative Navigation Solution," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 54, No. 3, pp. 189-205, 2007.

[29] Horn, J. P., "Bahnführung eines mobilen Roboters mittels absoluter Lagebestimmung durch Fusion von Entfernungsbild- und Koppelnavigations-daten," Ph.D. Dissertation, Technical University of Munich, 1997.

[30] Uijt de Haag, M., D. Venable, and M. Smearcheck "Integration of an Inertial Measurement Unit and 3D Imaging Sensor for Urban and Indoor Navigation of Unmanned Vehicles," in Proceedings of the Institute of Navigation National Technical Meeting 2007, pp. 829-840, January 2007.

[31] Soloviev, A., M. Uijt de Haag, “Three-Dimensional Navigation of Autonomous Vehicles Using Scanning Laser Radars: Concept and Initial Verification,” To appear in the *IEEE Transactions of Aerospace and Electronic Systems*, October 2009.

[32] Wahba, G., “A Least Squares Estimate of Satellite Attitude,” *SIAM Review*, Vol. 7, No. 3, p. 409, July 1965.

[33] Kabsch, W. “A Discussion of the Solution for the Best Rotation to Relate Two Sets of Vectors,” *Acta Crystallographica*, 34, 827-828, 1976.

[34] Mostov, K., A. Soloviev, and T. J. Koo, “Initial Attitude Determination and Correction of Gyro-Free INS Angular Orientation on the Basis of GPS Linear Navigation Parameters,” in *Proceedings of IEEE Conference on Intelligent Transportation Systems*, pp. 1034-1039, November 1997.

[35] M.L. Psiaki, “Attitude-Determination Filtering via Extended Quaternion Estimation,” Journal of Guidance, Control, and Dynamics, Vol. 23, No. 2, pp. 206-214, March-April 2000.