Pedestrian Tracking with Shoe-Mounted Inertial Sensors

Eric Foxlin
InterSense

A navigation system that tracks the location of a person on foot is useful for finding and rescuing firefighters or other emergency first responders, or for location-aware computing, personal navigation assistance, mobile 3D audio, and mixed or augmented reality applications. One of the main obstacles to the real-world deployment of location-sensitive wearable computing, including mixed reality (MR), is that current position-tracking technologies require an instrumented, marked, or premapped environment. Installing markers or instrumentation in advance is impractical for many mobile applications, and the hunt is on for a tracking method that will work reliably without preparation in any indoor or outdoor setting. Computer vision is the leading contender, but enormous challenges remain to developing a robust vision-based tracker for general-purpose use.

The NavShoe system uses a miniature inertial/magnetometer package wirelessly coupled to a PDA to provide navigation in arbitrary environments with and without GPS support.

A practical solution to orientation-only tracking is to use inertial sensors such as microelectromechanical systems (MEMS) gyroscopes with drift correction performed by referencing the earth's gravity for pitch and roll and the geomagnetic field for heading.[1] The self-contained sensors work in arbitrary unprepared indoor and outdoor environments. Unfortunately, no equally general solution exists for position tracking or localization, which MR systems require for registration. Until now, outdoor position tracking has had to rely on GPS or other radionavigation aids. Developers have suggested various acoustic, optical, or radio frequency (RF) localization systems for indoors, but all require installation of some infrastructure.

At InterSense, we've developed a system called NavShoe, which uses a new approach to position tracking based on inertial sensing. Our wireless inertial sensor is small enough to easily tuck into the shoelaces, and sufficiently low power to run all day on a small battery. Although it can't be used alone for precise registration of close-range objects, in outdoor applications augmenting distant objects, a user would barely notice the NavShoe's meter-level error combined with any error in the head's assumed location relative to the foot. Indoors, the system will still need computer vision to provide the precision tracking of the head-mounted display (HMD), but the approximate position information provided by NavShoe can greatly reduce the database search space for computer vision, making it much simpler and more robust.

The NavShoe device provides not only robust approximate position, but also an extremely accurate orientation tracker on the foot. The accuracy of heading direction measured by the NavShoe is much greater than what we can achieve using a head-mounted inertial orientation tracker because the higher foot accelerations enable the use of transfer alignment from GPS (see the "Transfer Alignment Explained" sidebar). Although few may be interested in a precision measurement of the heading of one of their feet, this foot-mounted device can serve as a portable azimuth reference for a handheld or head-mounted tracking sensor.

NavShoe concept

It's impossible to track position for more than a few seconds using inertial sensing alone.[2] Even a tiny drift rate in the gyros causes a slowly growing tilt error. The horizontal acceleration error is 9.8 m/s^2 times the tilt error in radians. Double integrating this increasing acceleration error produces a position error that grows cubically in time in the short term. Thus, although small inertial sensors can maintain accuracy of a few millimeters for one second, the drift will exceed a meter in 10 seconds. With NavShoe, we attempt to circumvent this t^3 drift problem by navigating open loop for only about 0.5 seconds at a time.

When a person walks, their feet alternate between a stationary stance phase and a moving stride phase, each lasting about 0.5 seconds. The NavShoe software detects the stance phase and applies zero-velocity updates (ZUPTs) as pseudomeasurements into the Extended Kalman Filter (EKF) navigation error corrector. This allows the EKF to correct the velocity error after each stride, breaking the cubic-in-time error growth and replacing it with an error accumulation that is linear in the number of steps.

Published by the IEEE Computer Society

Transfer Alignment Explained

The term "transfer alignment" originated with the problem of transferring initial alignment information (exact attitude of an inertial measurement unit, or IMU, relative to some geographical navigation coordinate frame) from a master IMU on an airplane to a slave IMU in an underwing munition, which must be quickly initialized in flight just before deployment. Because the wing might be flexing, the obvious approach of passing information about the aircraft's attitude to the slave IMU isn't accurate enough. Therefore, aviation engineers instead pass a sequence of position or velocity updates to the slave IMU, calculated based on the master IMU pose and known lever arm between them, and let the slave align itself using its own navigation filter. The alignment process works because any initial attitude error propagates into velocity errors when the IMU accelerates, as the off-diagonal term $\Delta t\mathbf{S}(\mathbf{f}^n)$ in the state transition matrix in the main article shows.

For example, imagine an IMU is at rest and level pointing in some yaw direction ψ, for example –90 degrees, but it thinks it has a yaw of $\psi + 5$ degrees. It accelerates in some direction, let's say due north, moves forward a distance R, and decelerates to a stop. Thus, the true final position is advanced by $[R\ 0\ 0]$ from the previous position in north-east-down navigation frame. The IMU will experience the northward acceleration/deceleration along its y-axis (which is pointed straight north), but because it thinks the y-axis is pointing +5 degrees east of north, it will calculate a final position advanced from the starting position by $[0.996R\ 0.087R\ 0]$. If it's then told, based on a string of GPS fixes, that the position actually advanced by about $[R\ 0\ 0]$, it will exploit the correlation between azimuth error and position error to correct its estimate of ψ back to –90 degrees. With NavShoe, each individual step R is only 1 to 2 meters, and the error in position increment caused by a 1-degree heading error is only a few centimeters, which is so small compared to the noise in a GPS fix that no significant amount of heading correction can occur. However, by the time the user walks 100 meters, the accumulated position error would be 1.7 meters, and dozens of GPS fixes will have been effectively averaged to produce enough information gain to begin reducing the heading error.

The accuracy and speed of attitude correction by transfer alignment depends on four main factors:

- frequency and magnitude of acceleration maneuvers,
- gyro drift rates,
- accelerometer precision and accuracy, and
- resolution of the GPS position fixes or velocity updates.

A microelectromechanical system IMU with 100-degree-per-hour gyro drifts and 1 mg accelerometer resolution would need to experience significant lateral accelerations (over 100 mg) frequently to develop the unambiguous correlations between heading error and position error needed for transfer alignment. This is why a shoe-mounted IMU can receive a much higher benefit from transfer alignment than an IMU on the torso can.

Introducing ZUPTs as measurements into the EKF instead of simply resetting the velocity to zero in the inertial integrator achieves important additional benefits. Most noticeably, the ZUPT lets the EKF retroactively correct most of the position drift that occurs during the stride phase. This is possible because the EKF tracks the growing correlations between the velocity and position errors in certain off-diagonal elements of the covariance matrix. For example, at the end of a stride, a high correlation between the uncertainty in north velocity and the newly accumulated uncertainty in northing position will exist. If the ZUPT indicates that the velocity error at the end of the stride was positive in the north direction, the EKF "knows" that it has been drifting north and will correct the position to the south and the velocity toward zero. The EKF is also able to correct pitch and roll error by taking advantage of the fact that tilt errors become correlated with horizontal velocity errors through the system dynamics matrix.

ZUPT pseudomeasurements let the EKF correct position, velocity, accelerometer biases, pitch, roll, and the pitch and roll gyro biases. Yaw (heading) and the yaw gyro bias are the only important EKF states that aren't observable from zero-velocity measurements. We've experimentally confirmed that operating the NavShoe with ZUPTs alone results in good short-term navigation performance but gradually loses horizontal position accuracy because of heading drift. If we plot the navigation results of a person walking in a large closed-loop rectangle in a map display, the tracked path appears to curve and doesn't return to the starting position.

We can solve the heading drift in several ways:

- use a much higher-performance gyro at least for the yaw axis, which exhibits negligible drift over the time period for which the NavShoe needs to be used;
- provide heading-correcting measurements from a magnetic compass; or
- provide position-correction measurements from GPS or other external aiding technologies.

The first method doesn't eliminate drift but strives to reduce the rate to an acceptable level. MEMS gyros currently perform at approximately 100 degrees per hour, and are expected to reach tactical-grade levels of around 1 degree per hour, although this could take another decade or longer. At this level, heading would drift only 0.3 degrees over a 20-minute mission, and the contribution to position error would be only a few meters per kilometer of travel. However, this article strives to demonstrate that we can achieve similar results today using commercial-grade MEMS gyros and a carefully calibrated and modeled solid-state magnetic compass (method 2). This approach will probably remain more cost-effective, compact, and power efficient for the foreseeable future.

The third method eliminates drift in the final computed position and indirectly corrects drift in yaw and yaw gyro bias through transfer alignment effects.

Previous Work in Pedestrian Dead-Reckoning Systems

Most previous personal dead-reckoning systems detect steps using a pedometer or accelerometer and move the position estimate forward by the step length in the direction determined by a magnetic compass or yaw gyro.[1-3] The waist- or torso-mounted sensors detect steps using the vertical component or the phasing of two or three acceleration axes exhibiting cycles typical of a human's walking motion. For a pedometer, which only counts steps, the step length is just the average for that user, perhaps adjusted for walking speed. More sophisticated systems analyze the accelerometer signals to estimate the magnitude of each step individually.

All of these systems require calibration to an individual user because everyone's gait has different acceleration profiles. Moreover, they always apply the step motion in the forward direction determined by the pelvis-mounted sensor. If a person steps in a different direction, their virtual position will still move forward. Some implementations[3] attempt to identify backward or sideways steps by the acceleration profiles, but they can never determine the exact direction of individual steps as precisely as the NavShoe does.

A 1996 DARPA project proposed using shoe-mounted inertial sensors with zero-velocity updating, but results were never published.[4] Other work describes an inertial navigation system embedded in a soldier's boot heel, but doesn't attempt experimental validation.[5] Stirling et al.[6] describe an experiment using a prototype shoe-mounted sensor that measures stride length with accelerometers and direction with magnetometers. Instead of gyros, their system measures angular acceleration using pairs of accelerometers. It also doesn't use a Kalman Filter to make optimal use of zero-velocity updates; the system simply stops integrating and resets the velocity before each step. Both features probably contribute to the unusable error of 10 to 20 percent of distance traveled.

To our knowledge, our implementation is the first to reap all of the benefits of shoe-mounted inertial navigation and achieve good performance with practical sensors.

References

1. T. Judd, "A Personal Dead Reckoning Module," *Proc. Inst. of Navigation Global Positioning System Conf.*, ION, 1997.
2. C. Randell, C. Djiallis, and H. Muller, "Personal Position Measurement Using Dead Reckoning," *Proc. 7th IEEE Int'l Symp. Wearable Computers,* (ISWC), IEEE CS Press, 2003, pp. 166-173.
3. M. Kourogi and T. Kurata, "Personal Positioning Based on Walking Locomotion Analysis with Self-Contained Sensors and a Wearable Camera," *Proc. 2nd IEEE and ACM Int'l Symp. Mixed and Augmented Reality* (ISMAR), 2003, pp. 103-112.
4. L. Sher, "Personal Inertial Navigation System (PINS)," DARPA, 1996; http://www.darpa.mil/MTO/SmartMod/Factsheets/Pins.html.
5. J. Elwell, "Inertial Navigation for the Urban Warrior," *Proc. SPIE*, vol. 3709, SPIE: The Int'l Soc. for Optical Eng., 1999, pp. 196-204.
6. R. Stirling et al., "An Innovative Shoe-Mounted Pedestrian Navigation System," *Proc. European Navigation Conf.* (GNSS), CD-ROM, Austrian Inst. of Navigation, 2003.

The "Previous Work in Pedestrian Dead-Reckoning Systems" sidebar discusses some other systems that use inertial sensors and/or magnetic compasses to keep track of pedestrian movements.

Implementation

For our initial proof-of-concept implementation, we combined some off-the-shelf hardware components with new algorithms developed in Matlab to postprocess the sensor data.

Hardware

Figure 1a diagrams the NavShoe system components. We mount only the InertiaCube3 and its battery on the foot. In the future, a PDA worn on a belt or vest will process the data. For the results presented here, we simply collected and postprocessed the data on a laptop. The GPS receiver's location on the body is not critical because GPS accuracy is only six meters. We used a Trimble LassenSQ receiver connected to the laptop USB port, and mounted the antenna on top of a baseball cap for maximum satellite visibility. The device is small (1 inch × 1 inch), low power, and low cost, and specifies 6-meter root-mean-square (RMS) accuracy using the wide-area augmentation system (WAAS).

The InterSense InertiaCube3, shown in Figure 1b, is a nine-element multisensor containing triaxialrate-gyros, accelerometers and magnetometers, and all associated analog and digital electronics in a 33 × 26 × 15-millimeter package. The nine sensors are factory calibrated and internally compensated to produce sensitive axes aligned with the package. Alignment and scale factor calibration residuals are on the order of 1 milliradian (mRad) and 3,000 parts per million (ppm), respectively.

One hurdle to making the NavShoe acceptable for real-world use was eliminating the cable running up the user's leg from the shoe-mounted sensor. This required adding an RF transceiver and cutting the power consumption to make the device battery friendly. The Wireless InertiaCube3 uses about half as much power as its InertiaCube2 predecessor—that is, 50 milliamperes (mA) at 5 volts (V)—and is almost half its size. The lower profile makes it easy to slip under the user's shoelaces. The radio supports 16 RF channels.

Software algorithms

Some key software algorithms enable the NavShoe to track position using inertial sensors and correct heading direction with a magnetic compass.

Inertial navigation and zero-velocity updating. NavShoe uses the standard strapdown inertial navigation system (INS) mechanization shown in Figure 2. The inertial measurement unit (IMU) reports compensated $\Delta\theta$ and $\Delta\mathbf{v}$ in body frame (hereafter b-frame) at 300 Hz. We use these reports to update the INS attitude, velocity, and position states, as the top two lines of the figure show. Drift correction is performed by a complementary EKF[3] operating on the error states $\delta\mathbf{x} = [\phi^T\ \delta\omega^T\ \delta\mathbf{r}^T\ \delta\mathbf{v}^T\ \delta\mathbf{a}^T]$,^{T} where ϕ^T represents attitude errors, $\delta\omega^T$ represents gyro biases, $\delta\mathbf{r}^T$ represents position errors, $\delta\mathbf{v}^T$ represents velocity errors, and $\delta\mathbf{a}^T$ represents accelerometer biases. After each estimation cycle, the filter delivers the error estimates to the INS, as the upward arrows show, and clears the error state vector $\delta\mathbf{x}$ back to zero. The state transition matrix for the discrete-time dynamics model is

$$\Phi = \begin{bmatrix} \mathbf{I} & -\Delta t\mathbf{C}_b^n & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I} & \Delta t\mathbf{I} & 0 \\ \Delta t\mathbf{S}\left(\mathbf{f}^n\right) & 0 & 0 & \mathbf{I} & \Delta t\mathbf{C}_b^n \\ 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix}$$

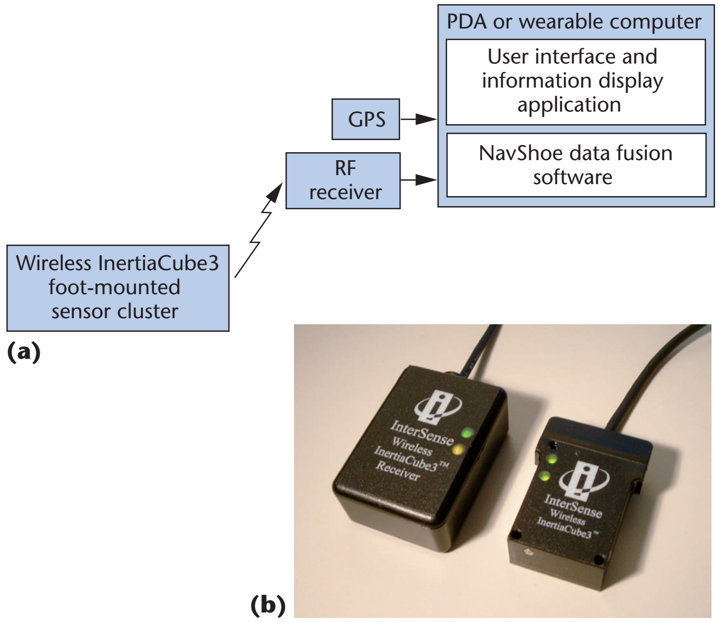
The rotation matrix $\mathbf{C}_b^n$ transforms vectors from b-frame to locally level navigation frame (n-frame) is included in the matrix subblocks (ϕ, $\delta\omega$) and ($\delta\mathbf{v}$, $\delta\mathbf{a}$) because the gyro and accelerometer biases are represented in b-frame. The expression $\mathbf{S}(\mathbf{f}^n)$ in the ($\delta\mathbf{v}$, ϕ) subblock represents the skew-symmetric cross-product-operator matrix

$$\begin{bmatrix} 0 & -f_z & f_y \\ f_z & 0 & -f_x \\ -f_y & f_x & 0 \end{bmatrix}$$

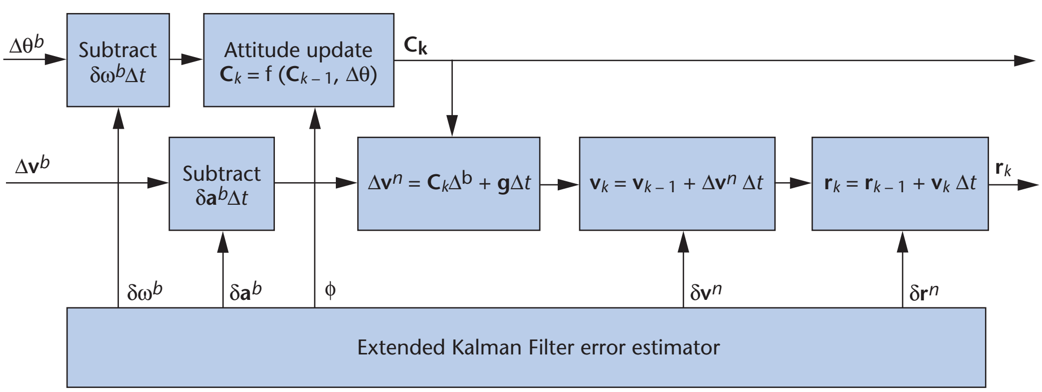
formed from the n-frame accelerometer output vector $\mathbf{f}^n = \mathbf{C}_b^n(\Delta\mathbf{v}^b/\Delta t)$. This term causes attitude errors to propagate into linear velocity errors in the n-frame, and thus lets the EKF build up the appropriate correlations in the covariance matrix to correct attitude errors based on measurements of n-frame velocity or position, as the "Transfer Alignment Explained" sidebar describes.

On each cycle, we propagate the covariance forward using $\mathbf{P}_{k+1|k} = \Phi_k \mathbf{P}_{k|k} \Phi_k^T + \mathbf{Q}_k$, where Φ_k is the above-defined state-transition-matrix, $\mathbf{P}_{k+1|k}$ is the estimation error covariance matrix at time $k+1$ based on measurements received through time k, and $\mathbf{Q}_k$ is the process noise covariance matrix, which is adaptively tuned to express the INS's single-cycle error growth due to sensor noise and calibration residuals.[3] We don't need to propagate the states with the time update equation $\delta\mathbf{x}_{k+1} = \Phi_k\delta\mathbf{x}_k$ because the EKF always transfers the error states to the INS and resets them to zero in the complementary filter immediately after a measurement.

If we make a measurement $\mathbf{z}$ relating to the underlying states $\mathbf{x}$ of the INS via the measurement equation $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v}$, where $\mathbf{v}$ is additive white zero-mean Gaussian noise with covariance matrix $\mathbf{R}$, and the current estimated states of the INS are $\hat{\mathbf{x}} = \mathbf{x} + \delta\mathbf{x}$, where $\mathbf{x}$ are the true states and $\delta\mathbf{x}$ are the estimation errors, then $\mathbf{z} = \mathbf{h}\left(\hat{\mathbf{x}} - \delta\mathbf{x}\right) + \mathbf{v} = \mathbf{h}\left(\hat{\mathbf{x}}\right) - \mathbf{H}\delta\mathbf{x} + \mathbf{v}$, where $\mathbf{H}$ is the Jacobian $\partial\mathbf{h}(\mathbf{x})/\partial\mathbf{x}$ of the possibly nonlinear measurement function $\mathbf{h}$. Therefore, the measurement model for the complementary error-state filter is $\nu = -\mathbf{H}\delta\mathbf{x} + \mathbf{v}$, where the INS innovation $\nu \equiv \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})$ serves as the measurement for the error-state filter, which is linear



1 The NavShoe system: (a) system block diagram and (b) Wireless InertiaCube3 (right) and its receiver (left).



2 Basic strapdown inertial navigation algorithm.

with measurement matrix $-\mathbf{H}$. After the EKF receives each measurement, it updates the state with $\delta\mathbf{x}_+ = \delta\mathbf{x}_- + \mathbf{K}(\nu - (-\mathbf{H})\,\delta\mathbf{x}_-) = \mathbf{K}\nu$, where the Kalman gain $\mathbf{K}$ is calculated with the usual formula: $\mathbf{K} = \mathbf{P}(-\mathbf{H})^T((-\mathbf{H})\mathbf{P}(-\mathbf{H})^T + \mathbf{R})^{-1} = -\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}$. It then updates the covariance using the Joseph form equation $\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}(-\mathbf{H}))\,\mathbf{P}_{k+1|k}(\mathbf{I} - \mathbf{K}(-\mathbf{H}))^T + \mathbf{K}\mathbf{R}\mathbf{K}^T$ where $\mathbf{I}$ represents the identity matrix.

In the current implementation, we perform the time update procedures on each integration step, 300 times per second, and select one of four types of measurement updates (ZUPT, magnetometer vector measurement, compass heading measurement, and GPS fix) depending on the system's mode. When the system is in magnetometer-calibration mode, we select from two measurement types (ZUPT and magnetometer vector measurement); when it's in navigation mode, we select from three (ZUPT, compass heading measurement, and GPS fix). If multiple measurement types are available, we select the type that was processed least recently. We perform the ZUPT if the zero-velocity algorithm (described in the next paragraph) determines that the foot is stationary. We assume a virtual measurement $\mathbf{z} = [0\ 0\ 0]^T$ of the INS velocity, for which the innovation ν is the negative of the current INS velocity estimate. The measurement matrix $\mathbf{H} = [0\ 0\ 0\ \mathrm{I}\ 0]$ selects the velocity error components (states 10, 11, and 12) from the error state vector $\delta\mathbf{x}$. We set the measurement noise covariance matrix to $\mathbf{R} = \mathbf{I} \times \max(\mathrm{trace}(\mathbf{H}\mathbf{P}\mathbf{H}^T), (1\text{ mm per second})^2)$. By making this matrix no smaller than the current velocity covariance, the ZUPT velocity reset occurs gradually over the half second that the foot is in contact with the ground, thus avoiding potential numerical instabilities with a large sudden reduction of covariance and problems that might occur if the system accidentally applied a total reset at a moment that the foot was not really still.

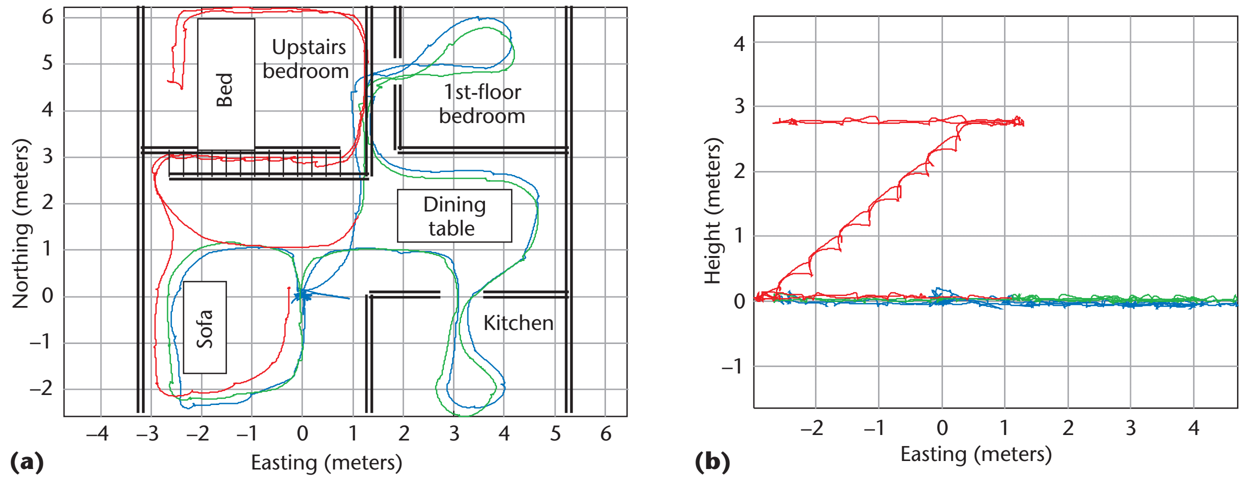
Accurately detecting the walking motion's stance phase, and thus the right time to apply ZUPTs, is rather easy. We increment the variable "stillTime" by Δt on each time step, resetting to zero if any of the gyro or accelerometer signals differs by more than a prescribed threshold from the average value of that signal since the current still period began. We tune each sensor's threshold to ensure that the still time is never more than 0.1 seconds during the stride phase, but almost always lasts for at least 0.3 seconds during the stance phase. The ZUPTing begins as soon as still time exceeds 0.15 seconds and continues as long as the foot remains still. This simple method worked well for all of the walking motions recorded. A more sophisticated approach might be required for running motions. Although running still times will be much shorter, we should be able to easily identify the beginning of the foot contact period by the sharp accelerometer spikes on heel strike.

Magnetometer calibration. Despite the precise factory calibration of the InertiaCube sensors, the compass didn't initially produce accurate results in the NavShoe application, as evidenced by large differences between the compass-measured yaw and gyro-predicted yaw. We assumed these errors were compass errors because they fluctuated faster than we expected the gyro to drift. We determined that one reason for this was the presence of metal in the shoe's sole, resulting in hard-iron biases and soft-iron distortions of the earth's magnetic field. To overcome this, we developed an in situ compass recalibration procedure to be performed after mounting the sensor on the shoe. The procedure augments the EKF with 13 additional states, representing the three hard-iron biases, nine soft-iron distortion parameters, and a horizontal magnetic field error to let the filter jointly estimate the local magnetic field inclination at the calibration site together with the 12 calibration parameters. During the calibration, the system employs a measurement model using the three measured magnetic field components from the magnetometers, with an $\mathbf{H}$ matrix containing partial derivatives of the magnetometer vector with respect to all 13 augmented states as well as the INS attitude states.

After inserting the sensor in the shoelaces, the user initiates the calibration procedure and wiggles the sensorized shoe for about 30 seconds, trying to rotate to many orientations while translating as little as possible so as to immerse the sensor in a fairly constant external magnetic field. When the user takes the first step with a translation of more than 0.5 meters, the calibration mode automatically terminates, and the system removes the 13 calibration parameters from the Kalman Filter. The system enters navigation mode and initializes the yaw covariance to an uncertainty of 10 degrees to accommodate the possibility that the horizontal field direction at the calibration site deviated from the local average direction. We don't need to calibrate the sensor again unless we remove it from the shoe and reinstall it in a different orientation or on a different shoe.

Geomagnetic modeling and heading drift correction. In navigation mode, the system doesn't use the raw magnetometer measurements directly, but uses them to compute a scalar compass-heading measurement. It compensates the raw magnetometer readings with the 12 magnetometer calibration parameters, then rotates them into the locally level n-frame using the INS pitch-and-roll estimates. The heading calculation uses the horizontal components of the transformed magnetometer vector, B_x and B_y, with the standard formula, heading = $\arctan2(B_y, B_x)$. The main advantage of using this preprocessed yaw measurement instead of the vector magnetic field measurement is that the yaw measurement is insensitive to the magnetic field's inclination and magnitude, and will always report heading with respect to the direction of the field's horizontal component only (assuming that highly accurate pitch-and-roll estimates are available from the INS for the transformation to the n-frame).

We introduce the *heading innovation* measurement (the difference between the compass-measured and the INS-predicted heading) to the error state filter using the 1×15 measurement matrix $\mathbf{H} = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, which relates the scalar heading error to the azimuthal attitude error $\phi(3)$. The measurement noise R is simply a scalar representing the compass measurement noise variance. However, we increase this noise to



3 **NavShoe trajectory during a 118.5-meter exploratory path through a house: (a) plan view and (b) elevation.**

include not only the white noise originating from the sensor, but also the colored measurement noises arising from magnetic disturbances in the environment. We model the compass output as, Compass reading = true heading + D (magnetic declination) + d (magnetic deviation) + sensor noise.

The magnetic declination, D, is the difference between magnetic north and true north arising from the tilt of the earth's magnetic field generator relative to the spin axis. The declination is a smooth global function represented by the International Geomagnetic Reference Field (IGRF) model of a few dipole field generators deep inside the earth, and doesn't include local anomalies caused by features near the surface. It varies plus or minus 20 degrees across the US, but changes little over a few kilometers. Thus, for NavShoe's purposes, we treat it as an unknown but constant offset throughout the mission.

The magnetic deviation d is the local disturbance of the horizontal field direction caused by local geological features, such as ore deposits, and cultural features, such as power lines, pipes, buildings, bridges, and vehicles. The value of d is constant at a given position but varies rapidly as a function of position, even over a scale of one meter for cultural features. Deviation size varies from a few degrees RMS outdoors, to tens of degrees inside an office building. In either case it is much larger than the sensor measurement noise, and it must be modeled carefully as the dominant additive noise in the yaw measurement process.

Originally, we tried to incorporate compass yaw measurements continuously, whether the foot with the NavShoe device was moving or still. However, this produces a difficult colored measurement noise problem, because all of the measurements taken when the foot is still are highly correlated (in fact almost the same), and thus will overly influence the INS heading estimate if treated as additive white measurement noise. One solution is to augment the Kalman Filter with a new state estimating the local deviation, d, modeled as a stochastic process with uncertainty growing rapidly during the stride phase. We decided it's simpler and more effective to take one compass yaw measurement per step. We take this measurement at the end of the stance phase, just as the foot begins to lift, because this is when pitch and roll have been corrected to their most accurate levels by a string of ZUPTs. Using this method, each yaw measurement exhibits a different error, which rapidly becomes decorrelated from previous compass errors if the magnetic deviation's spatial frequency is high. This is typically the case when cultural features cause most of the deviations. Large-scale magnetic anomalies caused by geological features such as ore deposits might pose a more difficult problem, but they're less common.

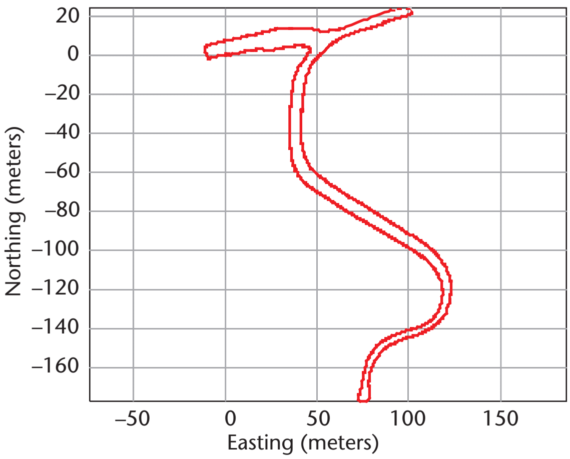
Test results

We performed indoor and outdoor navigation experiments using an unaided NavShoe—that is, inertial tracking with ZUPT and compass yaw updates, but no GPS.

Indoor experiment

In this experiment, a user walked through a typical wood-frame house for 322 seconds, covering a total distance of 118.5 meters. The user started at position (0, 0, 0) in the first-floor living room. The blue line in Figure 3a shows the first segment of the journey. The path loops around a sofa, back to the origin, into the kitchen, around the dining room table, into a first-floor bedroom, and back to the origin. A second loop along the same general path is shown in green. The user wasn't following a precise path on the floor, but tried to walk approximately the same course. As the map shows, the two traversals nearly coincide. The final segment (in red) goes left across the living room, then turns and climbs a staircase to the right, walks around the bed in the upstairs bedroom, then retraces itself back down the stairs, around the sofa, and back to the origin.

The tracker's final reported x, y, z position in meters is (–0.32, 0.10, –0.06), indicating that over the entire journey the position drifted by 0.3 percent of the distance traveled. As the elevation view in Figure 3b shows, NavShoe accurately tracks the height and can easily resolve each individual stair on the staircase. This is an important advantage over a simple pedometer and



4 NavShoe trajectory on a 741-meter road loop.

would allow a rescuer to unambiguously figure out in which room on which floor to look for a NavShoe-tracked firefighter. The total drift in altitude over the experiment was only 6 centimeters, or 0.06 percent of the distance traveled, consistent with our assertion that the main source of error is heading drift not quite corrected by the magnetic compass.

Outdoor experiment

Figure 4 shows the navigation performance of the unaided NavShoe outdoors. The user walked down a long winding hill on one side of a road and back up on the other, forming a closed loop with a total distance of 741 meters. We don't have reference information with which to evaluate the entire trip's absolute accuracy. However, the user ended the trip at the original starting point (0, 0, 0), and the NavShoe's reported position was displaced about 2 meters, indicating a probable accuracy of approximately 0.3 percent of distance traveled.

Integration with GPS

As our experiments show, the NavShoe alone can provide relative navigation accuracy of acceptable quality for a moderate distance, say up to a kilometer or two. For longer hikes, we can use it in conjunction with GPS to provide a hybrid GPS/inertial solution that's superior to ordinary GPS/INS, especially during extended GPS outages. Whereas an ordinary GPS/INS with a small MEMS IMU such as the InertiaCube3 can only coast for a few seconds before the position accuracy exceeds a few meters, the GPS/NavShoe system can coast for as long as it takes a person to walk several kilometers.

Aiding the NavShoe device with a GPS receiver also lets the system automatically identify the magnetic declination and put the results in a true geographic coordinate frame, as described in the next section. This is helpful in applications in which a user walks between indoor and outdoor areas. By letting GPS automatically calibrate the compass while the user is outdoors, the NavShoe can continue to represent the trajectory indoors in the same north-east-down coordinate frame without any user intervention.

Online calibration of magnetic declination

When GPS is available, we introduce it as a simple position-fix measurement into the NavShoe Kalman Filter with a noise covariance of $(20\ \text{m})^2 \times \mathbf{I}^{3\times3}$, which we increase several fold to account for high temporal correlations in the GPS errors. These position fixes bound the long-term position error to at least the GPS receiver's quality, and correct the yaw estimate (and thus indirectly the yaw gyro bias) through transfer alignment. Thus, during GPS availability, the system tracks the shoe-mounted sensor's yaw relative to true north with subdegree accuracy. To calibrate the local magnetic declination D and prepare the compass to take over the yaw drift correction when GPS transfer alignment is no longer available, we introduce a state for D into the Kalman Filter, and simultaneously accept yaw measurements from the compass according to the measurement model. We extend the measurement matrix $\mathbf{H}$ to include a partial derivative with respect to the new declination state as well as the true heading: $\mathbf{H} = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]$. Because we set the declination's initial uncertainty high (15 degrees RMS) to accommodate a completely unknown local D, the compass measurements initially have almost no effect on the NavShoe heading estimate, but serve instead to estimate D—basically, the average difference between the compass readings and the transfer-aligned yaw estimates in n-frame. After a user walks approximately 100 meters, the declination estimate converges to an uncertainty of about 0.5 degrees, and the compass information fuses with information from GPS for yaw drift correction. When GPS disappears, the compass continues to correct yaw in the same coordinate frame, which is now aligned to true north using the correct magnetic declination local value.

Test with simulated GPS outage

To test this integration strategy, we had a user walk a 1,059-meter closed loop through a hilly residential neighborhood, logging data from the NavShoe and GPS. During the first 400 meters, we incorporate the GPS fixes (shown in green in Figure 5) into the NavShoe Kalman Filter as described previously. For the remaining 659 meters, we ignore the GPS fixes (shown in blue) and plot them only for comparison. The NavShoe trajectory is in red, and conforms fairly closely to the GPS path during the training period. During the simulated GPS outage, the NavShoe track remains fairly closely aligned to the GPS track's direction, indicating that NavShoe learned the magnetic declination with good accuracy.

Significant discrepancies exist between the NavShoe position and the ignored GPS fixes, however. Whether the difference is due to NavShoe drift or GPS errors is difficult to say, but from casual inspection and prior knowledge that the roads don't suddenly change width, the NavShoe trajectory appears more reasonable than the GPS track. We hope that long-term navigation fusing GPS with NavShoe will produce more accurate results than GPS alone by filtering out temporary fluctuations in the GPS position such as the one near the top of Figure 5, where the GPS appears to temporarily drift about 10 meters north.

Other position-aiding approaches

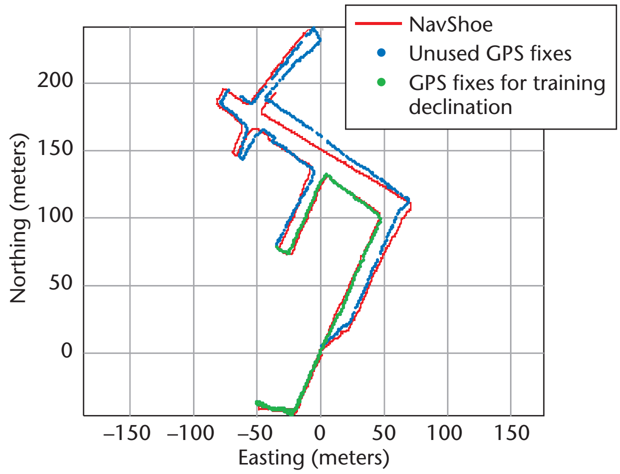
As we've shown, a user who walks into a building where no GPS is available can continue navigating in the same true geographical coordinate frame using NavShoe alone for a substantial distance—perhaps a kilometer or more depending on the accuracy requirement. Map-correlation techniques can let the user remain indoors longer if the floor plan is known. We simply find points where the user changes direction (according to NavShoe) and look for features in the floor plan (such as a doorway or corridor entrance) that could constrain where turns can occur. If we can correlate the turning behavior to a particular location on the floor plan with a high level of likelihood (for example, a person will likely turn to walk through a door), we can enter a position fix into the Kalman Filter as if it were a GPS fix. We can also implement more subtle constraints such as preventing the person from walking through a wall using 1D constraints instead of 2D or 3D fixes. If high accuracy is required, we could augment the NavShoe with sonar or optical ranging sensors to detect the NavShoe's distance from nearby walls. With a known floorplan, it's obvious how we could use this range information to update the position. However, even with no prior knowledge, we could construct a map on the fly using simultaneous localization and map-building (SLAM) algorithms from the robotics literature.

Buildings with wide-open spaces, where few opportunities for map correlation exist, might require an infrastructure such as an occasional beacon or sensor that provides information to NavShoe users as they walk by. Possibilities for implementing this include overhead cameras, photoelectric trip sensors, and encoded AC magnetic beacons that NavShoe magnetometers can detect and read.

Another position-aiding method that doesn't require an infrastructure is to use mutual ranging measurements between team members wearing NavShoe. We can do this with ultrasonic ranging or ultra-wideband (UWB) radio-ranging technologies, automatically initiating measurements between a pair of players when they come within range of each other and exchanging position estimates. This opens interesting questions in decentralized Kalman Filtering. A player shouldn't treat each range measurement as an independent piece of information and ignore the correlations of position errors between players that result from the shared measurement interactions.

One solution is to process the measurements using Covariance Intersection (CI) updates[4] instead of ordinary Kalman Filter updates. Another safe-but-conservative solution is to feed half of each measurement's information content to each player by duplicating the measurement while doubling each copy's noise covariance R. An optimal-but-expensive solution is to run a centralized Kalman Filter with states for all N players in each player's processor. The distributed information filtering literature offers many intriguing possibilities between these extremes.

Close-range MR applications require a more precise head position than NavShoe can provide. These applications typically perform head tracking with computer vision or hybrid vision/inertial tracking. The NavShoe can operate synergistically with a vision-based head-tracker. By providing a robust approximate initial-position guess, it drastically reduces the computer vision search problem's size. The vision system would only need to compare the feature descriptors in the current image to a small local subset of the feature database, which should make the initial data association much faster, especially if tracking relative to a large world model. In tracking mode, the vision system produces highly accurate head-position information, which we can use just like GPS fixes to correct the NavShoe.



5 NavShoe test integrating GPS, inertial navigation system (INS), and a compass and simulated GPS outage.

Heading transfer to head tracker or handheld device

Although long-range outdoor MR applications might find NavShoe's position-tracking precision sufficient, they require more accurate head orientation than a sourceless inertial head tracker can provide. Here again NavShoe can help.

The inertial head tracker ultimately depends on the magnetic compass for yaw drift correction, which means that it reports yaw rotation with respect to magnetic north instead of true north, and that the yaw accuracy will fluctuate with the magnetic deviations as the user walks.

The NavShoe can solve both of these problems because of its placement on the shoe. The frequent and large horizontal accelerations let it achieve much faster transfer alignment from GPS. This provides another source of yaw information that isn't affected by magnetic disturbances, and can achieve fraction-of-a-degree accuracy aligned to true north when GPS is available. Even when GPS drops out, the shoe-mounted compass, which has now been aligned to true north, continues to provide subdegree system accuracy despite much larger magnetic deviation errors in the compass output. The shoe-mounted compass can do this even when the same InertiaCube sensor on the head can't because the foot alternates between distinct phases of motion and stillness. This lets the NavShoe take exactly one compass measurement per step, which nearly decorrelates the compass measurements from each other. It also guarantees that if the user stops, there won't be a string of compass measurements corrupted by the same deviation, which gradually pulls the system heading off course. Conversely, the stopped condition lets the system use zero-angular-rate updates (ZARU), which freeze the system heading estimate in place and reset the yaw gyro bias.

The problem now is to impart the foot sensor's azimuth accuracy to the less accurate inertial sensor on the head. One possibility is to correlate the magnetic deviation at the ground level with the head-level deviation. The NavShoe would measure the instantaneous deviation (the difference between what the compass reports and its Kalman Filter-estimated system heading) and pass this information to the head sensor, which would compensate its compass. We'll investigate this simple solution in future work, but we suspect that in many environments, the correlation might not be tight enough to provide accurate compensation. If this is true, we might need an optical sensor to measure the head's yaw rotation relative to the foot. We might accomplish this by installing atop the foot a small laser or LED/lens combination that projects a line or a couple of dots on the ground in front of the foot, for example. A head-mounted camera angled toward the ground and equipped with a bandpass filter to block all wavelengths but that of the laser would see the pattern and measure head yaw with respect to it. This is more complex than the first solution, but much easier than developing a full-blown computer vision system for general outdoor environments.

Future work

Although the simple closed-loop tests suggest that the system accuracy is on the order of 0.3 percent, further experiments comparing the navigation results to a high-accuracy GPS reference will show whether this accuracy applies at all parts of the trajectory. Beyond this, our future plans are to integrate the tracker in a full MR system and try some of the suggested methods to increase the head-tracking sensor's yaw accuracy using the foot sensor as an attitude reference. The NavShoe is by no means intended to eliminate the need for computer vision in MR systems; however, it can increase the robustness of MR systems by continuing to provide position data when vision fails and can greatly reduce the burden on computer vision algorithm developers by providing an approximate starting position. ■

References

1. E. Foxlin, *Inertial Head-Tracking*, master's thesis, Dept. of Electrical Eng. and Computer Science, Massachusetts Inst. of Technology, 1993.
2. E. Foxlin, "Motion Tracking Technologies and Requirements," *Handbook of Virtual Environment Technologies*, K. Stanney, ed., Lawrence Erlbaum Publishers, 2002, chap. 8, p. 182.
3. E. Foxlin, "Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter," *Proc. Virtual Reality Ann. Int'l Symp.*, IEEE CS Press, 1996, pp. 185-194.
4. S. Julier and J. Uhlmann, "A Non-Divergent Estimation Algorithm in the Presence of Unknown Correlations," *Proc. Am. Control Conf.*, IEEE Press, 1997, pp. 2369-2373.

Eric Foxlin *is founder and chief technology officer at InterSense. His research interests include motion tracking, inertial navigation, sensor fusion, and environment mapping. Foxlin has a BA in physics from Harvard and an MS in electrical engineering and computer science from MIT. He's a member of the IEEE Computer Society and the Institute of Navigation.*