Development of Vision-aided Navigation for a Wearable Outdoor Augmented Reality System

Alberico Menozzi*, Brian Clipp†, Eric Wenger*, Jared Heinly‡,
Enrique Dunn‡, Herman Towles*, Jan-Michael Frahm‡, Gregory Welch§
*Applied Research Associates, Inc. (ARA), Raleigh, NC 27615 – Email: amenozzi@ara.com
†URC Ventures, Redmond, WA – Email: bclipp@gmail.com
‡University of North Carolina, Chapel Hill, NC – Email: jmf@cs.unc.edu
§University of Central Florida, Orlando, FL – Email: welch@ucf.edu

***Abstract*—This paper describes the development of vision-aided navigation (i.e., pose estimation) for a wearable augmented reality system operating in natural outdoor environments. This system combines a novel pose estimation capability, a helmet-mounted see-through display, and a wearable processing unit to accurately overlay geo-registered graphics on the user's view of reality. Accurate pose estimation is achieved through integration of inertial, magnetic, GPS, terrain elevation data, and computer-vision inputs. Specifically, a helmet-mounted forward-looking camera and custom computer vision algorithms are used to provide measurements of absolute orientation (i.e., orientation of the helmet with respect to the Earth). These orientation measurements, which leverage mountainous terrain horizon geometry and/or known landmarks, enable the system to achieve significant improvements in accuracy compared to GPS/INS solutions of similar size, weight, and power, and to operate robustly in the presence of magnetic disturbances. Recent field testing activities, across a variety of environments where these vision-based signals of opportunity are available, indicate that high accuracy (less than 10 mrad) in graphics geo-registration can be achieved. This paper presents the pose estimation process, the methods behind the generation of vision-based measurements, and representative experimental results.**

***Index Terms*—inertial navigation, augmented reality, computer vision.**

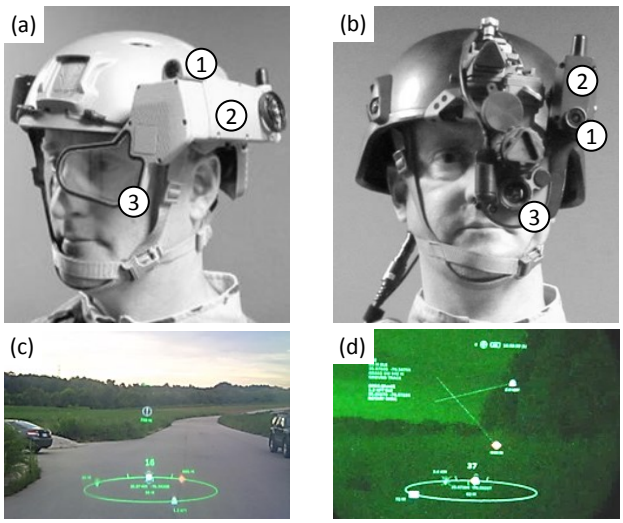

Fig. 1. Helmet-kit variants and corresponding augmented-reality (AR) view. Each helmet kit consists of a forward-looking camera (1), a sensor package (2), and a display (3). (See [2] for more details.) Image (a) shows the see-through display and (b) shows the night-vision display. Samples of their respective views are shown in images (c) and (d).

I. Introduction

In his 2009 review [1], Welch stated that the "Holy Grail" for researchers working on tracking for augmented reality (AR) "still seems to be robust and accurate tracking outdoors, for augmented reality everywhere. Researchers around the world are working on 6 DOF position and orientation-aware computer interfaces that will support access to information embedded in or attached to the physical world all around us." At around the same time, the effort described in this paper started, with the specific objective of developing a wearable AR system to provide intuitive visualization of geo-registered graphics on a see-through display. The core challenge has indeed been to achieve robust and accurate estimation of *pose* (i.e., position and orientation) outdoors.

In this particular application, the pose estimation problem is challenging on numerous fronts. The system must (a) track the pose of the user's head quickly and precisely (latency is very obvious when looking through a see-through display), (b) do so using relatively low-cost, low-SWAP (size, weight, and power) hardware in a ruggedized package, and (c) operate in arbitrary outdoor environments without requiring specific preparation or instrumentation. The overall system consists of a helmet kit (see Fig. 1), a processing unit, a graphical user interface (GUI) implementation, and a computer vision-aided navigation system. The system hardware and the GUI have been described in [2]. This paper provides additional details on the navigation system implementation.

The integration of inertial measurements with vision information has been the subject of a substantial amount of research and many approaches have been pursued [3]. The fundamental method that was chosen in this work consists of implementing a baseline GPS/INS and aiding it with vision-based information when it is available. The baseline GPS/INS is designed to provide a nominal level of performance when vision-aiding measurements are not available, and integrate them when available for improved performance. (No assumptions can be made about the presence or periodicity of vision-aiding measurements.) The level of performance without vision aiding is similar to that of off-the-shelf GPS/INS systems

978-1-4799-3320-4/14/$31.00 © 2014 IEEE

of similar SWAP, with additional measures to address latency and enhance robustness to magnetic and dynamic disturbances. When vision-based information is available, however, the current system is able to achieve a significant improvement in accuracy.

Though using vision-based information "all the time" was initially explored (e.g., using video frame-to-frame relative rotation/translation information [4], or using more sophisticated vision SLAM approaches [5], [6]), this path was not pursued further due to concerns about both processing power requirements and overall robustness. The vision algorithms currently in the system have instead been implemented as a module that may or may not provide measurements, depending on the circumstances. These are measurements of *absolute* orientation (i.e., orientation with respect to the Earth) that are generated by one or both of two vision-based methods: *landmark matching* (LM) and *horizon matching* (HM). Landmark matching requires the user to align a cross-hair (rendered on the display) with a distant feature of known coordinates, while horizon matching functions automatically without user involvement. Recent developments include the use of images of the Sun taken by the forward-looking camera. Corresponding *Sun-matching* (SM) absolute orientation measurements are also generated without user involvement.

The next section of this paper describes the overall method by discussing the pose estimation process and the methods behind the generation of vision-based measurements (including preliminary work on Sun matching). The remaining sections discuss experimental results of the integrated system and proposed future efforts.

II. Method

The main objective of AR is to render graphics on a display such that the graphical objects appear to be part of the real environment as the user looks through the display. This can be achieved if the position of any point in a reference coordinate system fixed to the environment can be also specified in the coordinate system of the display, which amounts to being able to accurately estimate the display's pose with respect to the environment.

Fig. 2 shows the various coordinate systems involved. The *body* coordinate system b is the reference for the helmet kit, with origin at the point p. The camera coordinate system c consists of a permutation of the body axes and shares the same origin. The display coordinate system d and the accelerometer coordinate system a are both rigidly attached to the body. (Coordinate system a is the reference for the helmet-kit's sensor package.) Coordinate system n is the North-East-Down (NED) reference for navigation. The Earth-Centered Earth-Fixed (ECEF) coordinate system e is used to specify points in the environment. Coordinate system i is the Earth-Centered Inertial (ECI) coordinate system, which is a good approximation of a true inertial reference in the context of this work. The WGS-84 ellipsoid [7] is used as the world model.

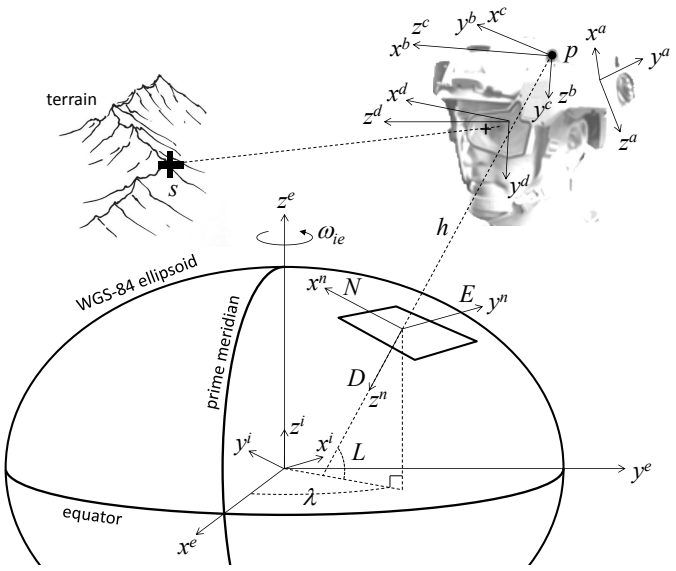

Fig. 2. Coordinate systems for pose estimation and rendering of geo-registered graphics.

Given the position $\boldsymbol{r}^e_{es}$ of a point s in the environment with respect the origin of e, expressed in e coordinates, its position with respect to the origin of d, expressed in display coordinates, can be computed as

$$\boldsymbol{r}^d_{ds} = \left(\mathsf{C}^e_n \mathsf{C}^n_b \mathsf{C}^b_d\right)^{\mathsf{T}} \left[\boldsymbol{r}^e_{es} - \left(\boldsymbol{r}^e_{ep} + \mathsf{C}^e_n \mathsf{C}^n_b \boldsymbol{r}^b_{pd}\right)\right], \qquad (1)$$

where $\boldsymbol{r}^e_{ep}$ is the position of p with respect to the origin of e, expressed in e coordinates, $\boldsymbol{r}^b_{pd}$ is the position of the origin of d with respect to p, expressed in b coordinates, and the C matrices represent orientation of one coordinate system (in the subscript) with respect to another (in the superscript). In practice, the position of points s and p with respect to the origin of e are specified in terms of latitude, L, longitude, λ, and altitude, h, in which case they are denoted as $\boldsymbol{p}_s$ and $\boldsymbol{p}_p$, respectively. The conversion from these geodetic coordinates to their Cartesian equivalent needed in (1) is performed by the mapping

$$\begin{aligned} x^e &= \left(R_N(L) + h\right) \cos L \cos \lambda \\ y^e &= \left(R_N(L) + h\right) \cos L \sin \lambda \\ z^e &= \left[\left(1 - \mathrm{e}^2\right) R_N(L) + h\right] \sin L \end{aligned} \quad , \qquad (2)$$

where $R_N(L)$ and e are WGS-84 ellipsoid parameters [7]. Since $\boldsymbol{p}_s$ is a given input, C^b_d and $\boldsymbol{r}^b_{pd}$ are obtained from a-priori calibration, and C^e_n is a known function of $\boldsymbol{p}_p$ [8, pg. 74], all that is needed to render graphics that are properly registered to the point s is accurate knowledge of $\boldsymbol{p}_p$ and C^n_b, which is the ultimate goal of the pose estimation process.

The remainder of this section describes the pose estimation process, the landmark matching and horizon matching methods, and the current development on the use of Sun-based measurements.

A. Pose Estimation Process

The pose estimation framework consists of an Extended Kalman Filter (EKF) implementation of a total-state loosely-coupled GPS/INS [8], designed to integrate additional aiding measurement of absolute orientation, without assumptions about their availability or periodicity. These "opportunistic" aiding measurements are provided by vision-based methods (LM and HM), which are described later. The main components and salient features of the pose estimation process are discussed below.

1) Calibration: Helmet-kit hardware calibration consists of estimating C_d^b, $\boldsymbol{r}_{pd}^b$, C_a^b, and $\boldsymbol{r}_{pa}^b$. Estimation of the relative orientation, C_a^b, of the sensor package with respect to the body is performed by following the procedure in [9], which also yields an estimate of the camera's intrinsic parameters. Estimation of the relative orientation, C_d^b, of the display with respect to the body is performed by an iterative process based on using the current C_d^b estimate to render scene features (e.g., edges) from camera imagery onto the display, and adjusting it until the rendered features align with the corresponding actual scene features when viewed through the display. The position vectors $\boldsymbol{r}_{pd}^b$ and $\boldsymbol{r}_{pa}^b$ can be obtained by straightforward measurement, but in fact they are negligible in the context of this application (the former because $\|\boldsymbol{r}_{pd}\| \ll \|\boldsymbol{r}_{ps}\|$, and the latter because its magnitude is very small and was empirically determined to have negligible effect). The magnetometer is calibrated prior to each operation by following the procedure described in [10].

2) EKF Models: The EKF is based on the model

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, t) \\ \hat{\boldsymbol{y}}_k &= \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{\nu}_k) \end{aligned},$$

where t is time, $\boldsymbol{f}$ is the continuous-time process, $\boldsymbol{h}_k$ is the discrete-time measurement (with output $\hat{\boldsymbol{y}}_k$), $\boldsymbol{x}$ is the state vector, $\boldsymbol{x}_k$ is its discrete-time equivalent, and $\boldsymbol{u}$ is the input vector. The vector $\boldsymbol{w}$ is a continuous-time zero-mean white-noise process with covariance Q (denoted as $\boldsymbol{w} \sim (\boldsymbol{0}, \mathsf{Q})$), and $\boldsymbol{\nu}_k$ is a discrete-time zero-mean white-noise process with covariance R_k (denoted as $\boldsymbol{\nu}_k \sim (\boldsymbol{0}, \mathsf{R}_k)$).

The state is defined as $\boldsymbol{x} = \left[\boldsymbol{p}_p; \boldsymbol{v}_{ep}^n; \boldsymbol{q}_{nb}; \boldsymbol{b}_g; \boldsymbol{b}_a; b_\alpha; b_\gamma\right]$ (semicolons are used as row separators), where $\boldsymbol{v}_{ep}^n$ is the velocity of the point p with respect to the ECEF coordinate system, expressed in NED coordinates, and $\boldsymbol{q}_{nb}$ is the quaternion representation of C_b^n. The vector $\boldsymbol{b}_g$ is the rate gyro bias, $\boldsymbol{b}_a$ is the accelerometer bias, and b_α and b_γ are biases in the model of local magnetic declination and inclination values, respectively.

The rate gyro and accelerometer data are inputs to the process model, so that $\boldsymbol{u} = [\boldsymbol{u}_a; \boldsymbol{u}_g]$, with

$$\begin{aligned} \boldsymbol{u}_a &= \boldsymbol{f}_{ip}^b + \boldsymbol{b}_a + \boldsymbol{w}_a \\ \boldsymbol{u}_g &= \boldsymbol{\omega}_{ib}^b + \boldsymbol{b}_g + \boldsymbol{w}_g \end{aligned},$$

where $\boldsymbol{f}_{ip}^b = \mathsf{C}_b^{n\mathsf{T}}\left[\boldsymbol{a}_{ep}^n - \boldsymbol{g}^n + (\boldsymbol{\omega}_{en}^n + 2\boldsymbol{\omega}_{ie}^n) \times \boldsymbol{v}_{ep}^n\right]$ is the *specific force* at p, $\boldsymbol{\omega}_{ib}^b$ is the angular rate of the body coordinate system with respect to the ECI coordinate system, $\boldsymbol{w}_a \sim (\boldsymbol{0}, \mathsf{Q}_a)$, and $\boldsymbol{w}_g \sim (\boldsymbol{0}, \mathsf{Q}_g)$. The cross product in the $\boldsymbol{f}_{ip}^b$ expression is a Coriolis and centripetal acceleration term due to motion over the Earth's surface [11], and can be neglected when the velocity is small (which is the case for pedestrian navigation).

Using the state definition and input model described above, the process model is specified by the following equations:

$$\begin{aligned} \dot{\boldsymbol{p}}_p &= \boldsymbol{f}_p(\boldsymbol{x}) + \boldsymbol{w}_p \\ \dot{\boldsymbol{v}}_{ep}^n &= \mathsf{C}_b^n(\boldsymbol{u}_a - \boldsymbol{b}_a - \boldsymbol{w}_a) + \\ &\quad + \boldsymbol{g}^n - (\boldsymbol{\omega}_{en}^n + 2\boldsymbol{\omega}_{ie}^n) \times \boldsymbol{v}_{ep}^n + \boldsymbol{w}_v \\ \dot{\boldsymbol{q}}_{nb} &= \frac{1}{2}\Omega(\boldsymbol{q}_{nb})\left(\boldsymbol{u}_g - \boldsymbol{b}_g - \boldsymbol{w}_g - \boldsymbol{\omega}_{in}^b\right) + \boldsymbol{w}_q \\ \dot{\boldsymbol{b}}_g &= \boldsymbol{w}_{b_g} \\ \dot{\boldsymbol{b}}_a &= \boldsymbol{w}_{b_a} \\ \dot{b}_\alpha &= w_\alpha \\ \dot{b}_\gamma &= w_\gamma \end{aligned}$$

where $\boldsymbol{f}_p$ is a known function of $\boldsymbol{v}_{ep}^n$, h, L, and WGS-84 parameters [8, pg. 61], $\boldsymbol{g}^n$ is the acceleration due to gravity, Ω is a 4×3 matrix that transforms an angular rate vector into the corresponding quaternion derivative [11, pg. 44], and $\boldsymbol{\omega}_{in}^b = \mathsf{C}_b^{n\mathsf{T}}(\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n)$. The process noise vector is $\boldsymbol{w} = \left[\boldsymbol{w}_p; \boldsymbol{w}_v; \boldsymbol{w}_q; \boldsymbol{w}_g; \boldsymbol{w}_{b_g}; \boldsymbol{w}_a; \boldsymbol{w}_{b_a}; w_\alpha; w_\gamma\right]$, and $\mathsf{Q} = \mathrm{blkdiag}\left(\mathsf{Q}_p, \mathsf{Q}_v, \mathsf{Q}_q, \mathsf{Q}_g, \mathsf{Q}_{b_g}, \mathsf{Q}_a, \mathsf{Q}_{b_a}, \sigma_\alpha^2, \sigma_\gamma^2\right)$ is its covariance matrix.

The measurement vector is defined as

$$\hat{\boldsymbol{y}}_k = \begin{bmatrix} \hat{\boldsymbol{y}}_{\mathrm{LM}} \\ \hat{\boldsymbol{y}}_{\mathrm{HM}} \\ \hat{\boldsymbol{y}}_a \\ \hat{\boldsymbol{y}}_m \\ \hat{\boldsymbol{y}}_{\mathrm{Gv}} \\ \hat{\boldsymbol{y}}_{\mathrm{Gp}} \\ \hat{y}_{\mathrm{D}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_{nb} + \boldsymbol{\nu}_{\mathrm{LM}} \\ \boldsymbol{q}_{nb} + \boldsymbol{\nu}_{\mathrm{HM}} \\ \mathsf{C}_b^{n\mathsf{T}}\left(\boldsymbol{a}_{ep}^n - \boldsymbol{g}^n\right) + \boldsymbol{b}_a + \boldsymbol{\nu}_a \\ \mathsf{C}_b^{n\mathsf{T}}\boldsymbol{m}^n + \boldsymbol{\nu}_m \\ \boldsymbol{v}_{ep}^n + \boldsymbol{\nu}_{\mathrm{Gv}} \\ \boldsymbol{p}_p + \boldsymbol{\nu}_{\mathrm{Gp}} \\ h + \nu_{\mathrm{D}} \end{bmatrix}, \quad (3)$$

where landmark matching, $\hat{\boldsymbol{y}}_{\mathrm{LM}}$, and horizon matching, $\hat{\boldsymbol{y}}_{\mathrm{HM}}$, are the vision-aiding measurements, $\hat{\boldsymbol{y}}_a$ is the accelerometer measurement, $\hat{\boldsymbol{y}}_m$ is the magnetometer measurement, $\hat{\boldsymbol{y}}_{\mathrm{Gv}}$ is the velocity measurement, $\hat{\boldsymbol{y}}_{\mathrm{Gp}}$ is the GPS horizontal position (i.e., latitude and longitude) measurement, and $\hat{y}_{\mathrm{D}}$ is the measurement of altitude based on Digital Terrain and Elevation Data (DTED). The measurement noise vector is $\boldsymbol{\nu}_k = [\boldsymbol{\nu}_{\mathrm{LM}}; \boldsymbol{\nu}_{\mathrm{HM}}; \boldsymbol{\nu}_a; \boldsymbol{\nu}_m; \boldsymbol{\nu}_{\mathrm{Gv}}; \boldsymbol{\nu}_{\mathrm{Gp}}; \nu_{\mathrm{D}}]$, and its covariance matrix is $\mathsf{R}_k = \mathrm{blkdiag}\left(\mathsf{R}_{\mathrm{LM}}, \mathsf{R}_{\mathrm{HM}}, \mathsf{R}_a, \mathsf{R}_m, \mathsf{R}_{\mathrm{Gv}}, \mathsf{R}_{\mathrm{Gp}}, \sigma_{\mathrm{D}}^2\right)$. Because of the block-diagonal structure of R_k, the EKF measurement update step is executed by processing measurements from each sensor as separate sequential updates (in the same order as they are listed in (3)).

The gravity vector is approximated as being perpendicular to the ellipsoid and therefore modeled as $\boldsymbol{g}^n = [0; 0; g_0(L)]$, where the down component $g_0(L)$ is obtained from the Somigliana model [7]. Note that since the acceleration $\boldsymbol{a}_{ep}^n$ is not directly measured nor modeled (accelerometers can only measure specific force), it appears in (3) as an unknown

quantity that, when nonzero, has the effect of degrading the value of accelerometer measurements in aiding the estimate of orientation.

The reference magnetic field vector $\boldsymbol{m}^n$ in (3) is the Earth's magnetic field vector, expressed in n coordinates, and is modeled as

$$\boldsymbol{m}^n = \begin{bmatrix} \cos(\hat{\alpha} - b_\alpha)\cos(\hat{\gamma} - b_\gamma) \\ \sin(\hat{\alpha} - b_\alpha)\cos(\hat{\gamma} - b_\gamma) \\ \sin(\hat{\gamma} - b_\gamma) \end{bmatrix} B_m, \qquad (4)$$

where B_m is the Earth's magnetic field strength, and $\hat{\alpha}$ and $\hat{\gamma}$ are the values of magnetic declination and inclination, respectively, obtained from the EMM2010 Earth magnetic model [12]. Because they are otherwise not observable, updating of the corresponding biases, b_α and b_γ, is only allowed when a vision-aiding measurement is available.

3) Initialization and Alignment: The initial state $\boldsymbol{x}(0)$ is estimated by using sensor readings during the first few seconds of operation before the EKF process starts. The initial condition of all biases is set to zero.

4) Parameter Tuning: Parameter tuning consists of establishing values for Q, R_k, the initial estimated error covariance matrix $\mathsf{P}(0)$, and a number of parameters that are used for disturbance detection, filtering, etc. This tuning has been performed by combining Allan variance analysis of sensor data [13], with the models described so far, to identify a starting point. Further adjustments were performed based on experiments.

5) Dynamic Disturbance: Since they are used as measurements of the gravity vector in body coordinates, accelerometer-based updates are only valid if $\boldsymbol{a}^n_{ep}$ is zero (see (3)). If not, these measurements are considered to be corrupted by an unknown *dynamic disturbance*. The problem is addressed by detecting the presence of this disturbance and, if detected, increasing the corresponding measurement noise covariance matrix, R_a, by a large factor ρ_a. Detection is based on comparing the norm of the accelerometer measurement to $\|\boldsymbol{g}^n\|$, as well as checking that the measured angular rate is low enough. (The location of the sensor package on the helmet kit, and the corresponding kinematics, result in angular rate being a very good indicator of $\boldsymbol{a}^n_{ep}$.) The approach of increasing R_a implies that the unknown acceleration $\boldsymbol{a}^n_{ep}$ is modeled as a stationary white noise process. Though the actual process is not stationary nor white, it was found experimentally that this approach yields better results than the alternative of completely rejecting accelerometer measurements that are deemed disturbed. In fact, when testing this alternative, it was observed that a single valid measurement after long periods of dynamic disturbance (as is the case when walking) could cause undesirable jumps in the estimates of $\boldsymbol{b}_g$ and $\boldsymbol{b}_a$, while increasing R_a resulted in no such issues.

6) Magnetic Disturbance: Magnetometer-based measurement updates are valid if the magnetic field being measured is the Earth's magnetic field only. Otherwise, these measurements are considered to be corrupted by an unknown *magnetic disturbance*. The problem is addressed by detecting the presence of magnetic disturbances and, if detected, rejecting the corresponding magnetometer measurements. Detection is based on comparing the norm of the measured magnetic field vector to the Earth's field strength B_m, as well as checking that the computed inclination angle is not too far from the nominal value. (Since it is based on $\boldsymbol{y}_m^{\mathsf{T}}\boldsymbol{y}_a$, the latter check is only performed if no dynamic disturbance is detected.)

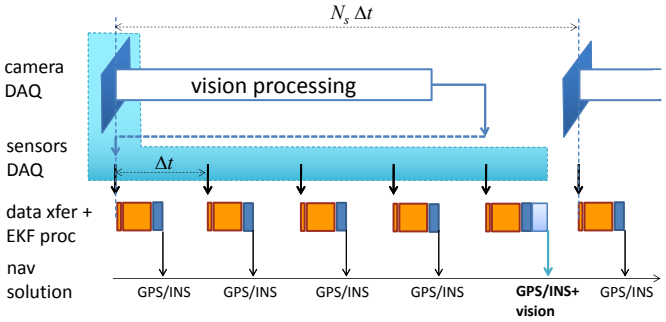

Fig. 3. Qualitative timing diagram of vision and EKF processing. When vision-based information is processed and delivered, the EKF must reprocess past information. This extra processing is handled within a single EKF epoch Δt. Note that the image acquisition is synchronized with the sensor data acquisition.

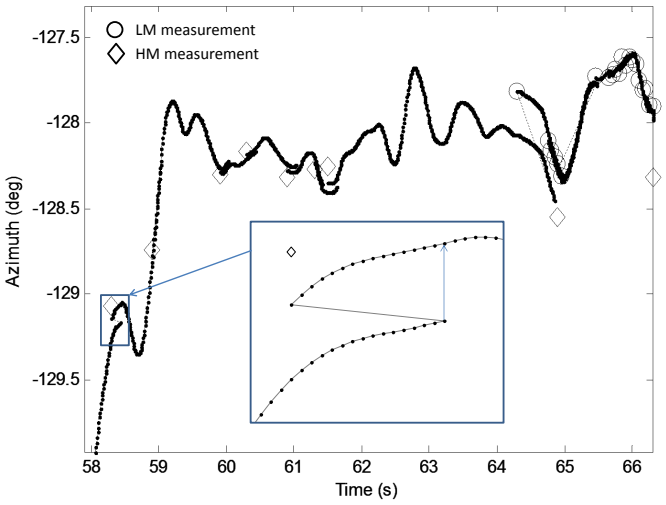

Fig. 4. Vision-aiding measurement update. In the example illustrated by the inset, the EKF is able to "go back in time" and use the rewind buffer to reprocess the azimuth estimate based on the delayed HM measurement.

7) The Rewind Buffer: The rewind buffer (RB) is a circular buffer that is used to maintain a record of relevant information pertaining to the last N_r samples of EKF processing. This is done to properly integrate vision-aiding measurements, which are delayed with respect to the rest of the data (see Fig. 3). Fig. 4 shows a close-up look at an azimuth update based on a horizon matching measurement. The EKF is able to "go back in time" and reprocess the state estimate based on the late measurement, all within its regular processing interval.

8) Altitude: In this effort, DTED has been used as the only measurement of altitude. Since DTED is readily available and experiments have taken place almost exclusively on natural

terrain, other options have not yet been explored. However, planned future tasks include integrating GPS and barometric altitude measurements.

9) The Forward Buffer: The forward buffer (FB) is a buffer that is used to store both the current state estimate $\boldsymbol{x}_k^+$ and the predicted state estimates up to N_f time steps ahead. That is, $\mathrm{FB}_k = \left\{\boldsymbol{x}_k^+, \boldsymbol{x}_{k+1}^-, \boldsymbol{x}_{k+2}^-, \ldots, \boldsymbol{x}_{k+N_f}^-\right\}$. Through interpolation of the FB vectors, a state estimate can then be produced for any $t \in [t_k, t_k + N_f\Delta t]$, where t_k is the time of the current estimate and Δt is the EKF's processing interval. Given a value, Δt_d, for system latency, the pose that is delivered at time t_k for rendering graphics on the display is based on the predicted state at $t = t_k + \Delta t_d$, which is extracted from the FB. (Note that N_f must be selected such that $N_f > 0$ and $N_f\Delta t \geq \Delta t_d$.) The beneficial effect of this forward-prediction process is obvious when using the system, and focused experiments [2] have shown a reduction in perceived latency from about 40 ms to about 2 ms.

10) Adaptive Gyro Filtering: The forward-prediction process extrapolates motion to predict the state at some time in the future, and is inherently sensitive to noise. This may result in jitter of the rendered graphics even when the system is perfectly stationary (e.g., mounted on a sturdy tripod). Low-pass filtering of the rate gyro signal, $\boldsymbol{u}_g$, reduces this jitter effect but also introduces lag. Since this lag is not noticeable when the rotation rate is near zero, and the jitter is not noticeable when there is actual motion, a reduction in perceived jitter is achieved by low-pass filtering the rate gyro signal only when the estimated rotation rate magnitude is small. This can be done by adjusting the low-pass filter's bandwidth using a smooth increasing function of estimated rotation rate magnitude. The resulting filtered signal can be then used in place of $\boldsymbol{u}_g$ in the EKF's time-propagation steps (i.e., in the forward-prediction process). This method was found to reduce jitter by a factor of three without any adverse effects in other performance measures [2].

11) Pose Estimation Processing Step: A single pose estimation processing step takes as inputs the current sensor data, the RB data, and an index, i_{now}, corresponding to the current-time location in the RB. It returns updates to RB, i_{now}, and the whole FB. It is implemented as follows:

1: pre-process sensor data
2: $\mathrm{RB}[i_{\mathrm{now}}] \leftarrow$ {sensor data, pre-processed data}
3: $i_{\mathrm{stop}} = i_{\mathrm{now}}$
4: **if** vision data is available **and** $\exists\, i_{\mathrm{vis}} : t_{\mathrm{CLK}}$ in $\mathrm{RB}[i_{\mathrm{vis}}]$ = t_{CLK} in vision data **then**
5: $\quad i_{\mathrm{now}} = i_{\mathrm{vis}}$
6: **end if**
7: `keep_processing` = **true**
8: **while** `keep_processing` = **true do**
9: $\quad \{\boldsymbol{x}^-, \mathsf{P}^-\} \leftarrow \mathrm{RB}[i_{\mathrm{now}}]$
10: $\quad \mathrm{RB}[i_{\mathrm{now}}] \leftarrow \{\boldsymbol{x}^+, \mathsf{P}^+\} =$ `ekf_u`$(\boldsymbol{x}^-, \mathsf{P}^-, \mathrm{RB}[i_{\mathrm{now}}])$
11: $\quad i_{\mathrm{next}} = i_{\mathrm{now}} + 1$
12: $\quad \mathrm{RB}[i_{\mathrm{next}}] \leftarrow \{\boldsymbol{x}^-, \mathsf{P}^-\} =$ `ekf_p`$(\boldsymbol{x}^+, \mathsf{P}^+, \mathrm{RB}[i_{\mathrm{now}}])$
13: $\quad$ **if** $i_{\mathrm{now}} = i_{\mathrm{stop}}$ **then**
14: $\quad\quad \mathrm{FB}[0] \leftarrow \boldsymbol{x}^+, \mathrm{FB}[1] \leftarrow \boldsymbol{x}^-$
15: $\quad\quad$ **for** $k_p = 2$ **to** N_f **do**
16: $\quad\quad\quad \{\boldsymbol{x}^-, \mathsf{P}^-\} =$ `ekf_p`$(\boldsymbol{x}^-, \mathsf{P}^-, \mathrm{RB}[i_{\mathrm{now}}])$
17: $\quad\quad\quad \mathrm{FB}[k_p] \leftarrow \boldsymbol{x}^-$
18: $\quad\quad$ **end for**
19: $\quad\quad$ `keep_processing` = **false**
20: $\quad$ **end if**
21: $\quad i_{\mathrm{now}} = i_{\mathrm{next}}$
22: **end while**

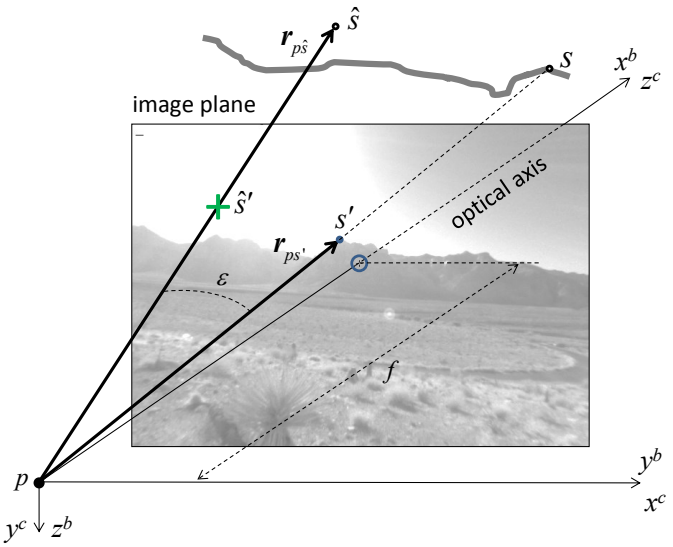

Fig. 5. Camera model and vectors used to measure accuracy.

where t_{CLK} is the reference time stamp of both sensor and vision data acquisition. Lines 10 and 12 are the EKF measurement update and prediction steps, respectively. The loop on lines 15–18 implements the forward-prediction process by repeating single EKF prediction steps.

12) Error Metric: Accuracy performance is based on a measure of error, ϵ, defined as the angle between the vectors $\boldsymbol{r}^b_{ps'}$ and $\boldsymbol{r}^b_{p\hat{s}}$ (see Fig. 5). The point s' is the point in the undistorted camera image corresponding to the real-world reference point s, and is obtained via semi-automatic processing (i.e., requiring some manual input) of the imagery. The vector $\boldsymbol{r}^b_{p\hat{s}}$ is the result of using the pose estimate, $\{\boldsymbol{p}_p, \mathsf{C}^n_b\}$, to compute $\boldsymbol{r}^b_{ps}$. Note that, in addition to pose estimation errors, the process of generating the "ground-truth" vector $\boldsymbol{r}^b_{ps'}$ also contributes to ϵ. This contribution was observed to be up to 3 mrad across a variety of experiments.

B. Landmark Matching

The landmark matching (LM) module uses imagery from the forward-looking camera to track the location of a distant object of known coordinates and provide a measurement of orientation to the EKF. Prior to operation, the user must select a feature in the environment (i.e., a landmark) that can be visually recognized during operation and whose coordinates are known. Once in the area of operation, the user overlays

a cross hair — rendered on the display and corresponding to the intersection of the camera's optical axis with the image plane (shown as a circle in Fig. 5) — on the selected landmark and clicks a mouse button. This procedure is called "landmark clicking".

1) LM Method: Landmark clicking triggers the system to extract features from the current image and compute the corresponding absolute orientation of the camera (and therefore the body) using the known direction of the optical axis and the EKF's current estimate of roll angle. The combination of extracted features and absolute orientation is stored in a *landmark key-frame* that can be compared to later images to determine their corresponding camera orientations. Fig. 6 shows an illustration of the LM process in action. Once the landmark key-frame is generated by the user, the LM module uses computer vision techniques to determine orientation.

Regarding the extraction of features in a given image, FAST [14] corners are extracted in the undistorted image and their BRIEF [15] descriptors are calculated. The tilt estimate from the EKF is then used to align the BRIEF descriptors with respect to the down axis of the NED coordinate system. This eliminates the need for rotational invariance and increases the discrimination power of the BRIEF descriptors compared to feature descriptors, such as Oriented BRIEF (ORB) [16], that use image gradient information to orient the descriptors.

It is important to maintain robustness to the user walking short distances where the landmark is still in view after moving. Therefore, nearby image features, which move due to parallax as the user walks, must be separated from far features, which do not move. This can be done by a model-fitting approach consisting of fitting either an essential matrix [17], in the case where features are close, or a rotation matrix when all of the features are far away. In practice, it was found that in most cases features at intermediate distances exhibited a small degree of parallax yet still fit a rotation-only hypothesis model within the required accuracy. The small parallax in these features, however, was enough to create a bias in the rotation estimate and caused a corresponding orientation error to be passed on to the EKF.

To alleviate this issue, a simple heuristic approach to feature selection was implemented, based on choosing only features that are above a threshold distance from the camera. This distance is computed using the EKF's tilt estimate and the assumption of a flat ground in front of the camera. Ultimately, robustness of LM to translation depends on the user being trained to use it only for distant landmarks, without nearby objects in the scene to cause parallax. (Of course, the ideal situation for the current LM implementation is one where the camera is not translating at all.)

After extraction, features in the current image are matched to features in the landmark key-frame based on their BRIEF descriptors, calculating the best matching feature as the one with minimum Hamming distance. For each feature in the landmark key-frame, its best match in the current image is computed. The same is done from the current image to the landmark key-frame and only those matches that agree in both directions are deemed valid. (This is a standard process of cross-validation of feature matches.) After matching, a two-point RANSAC [18] procedure is applied to find the rotation between the two frames and eliminate outliers. Because the camera has been calibrated, only the three degrees of freedom of the relative rotation between the landmark key-frame and current images need to be estimated. Two feature matches provide four constraints and so over-constrain the solution. Each potential rotation solution is scored in the RANSAC procedure by rotating the current image's features according to the inverse of the rotation and applying a threshold to the distance to the corresponding feature matches in the landmark frame. The number of feature matches satisfying the threshold is the score for that solution.

Before delivering a measurement of orientation to the EKF, a few sanity checks must be satisfied. At least M features matches are required between the landmark key-frame and the current frame after RANSAC. This prevents incorrect rotations with little support in the features from being passed to the EKF. The RANSAC procedure must also exceed a minimum target confidence in its solution. This confidence is calculated as the probability $p = 1-(1-i^s)^n$, where n is the number of RANSAC iterations, s is the number of points selected at each iteration, and i is the inlier ratio. (A lower bound of the true inlier ratio can be computed by dividing the maximum number of inliers that was observed by the total number of feature matches.) An upper bound on n is set to limit processing time and meet real-time constraints, so it is possible that p may not reach the required level. The inlier ratios observed in practice and the small number of points selected (i.e., $s = 2$) result in a high-enough p most of the time. A final check is that the angle between the optical axis of the landmark key-frame and that of the current frame be less than twenty degrees, insuring adequate overlap between the two images.

A key feature of the LM method is that the object needs to be visible to the user but not necessarily to the camera. Since the LM module tracks FAST corner features around the landmark object, these features need not be on the landmark object itself.

2) Standalone LM Results: Standalone results are produced by using the LM orientation measurement, $\boldsymbol{y}_{\mathrm{LM}}$, to compute ϵ, instead of using the EKF's orientation estimate. The LM module has been tested in a variety of environments to verify the generality of its algorithms. One example is an area of the Smoky Mountains (see images in Fig. 9), characterized by green, tree-covered, rounded mountains, as well as plenty of vegetation in the lower flatter areas. Another example is the area around Red Rock Canyon, NV (see images in Fig. 6), characterized by desert-like landscape, with rocks and bushes in the flat areas, and sharp, bare-rock mountain peaks. A sample of performance of the LM module in these environments is shown in plots (a) of Fig. 7 and Fig. 8. LM performed well in both cases, with a mean error less than 3 mrad. Because the landmark is designated by the user, and can only be matched when it is in the field of view of the camera, only limited sections of the data have landmark

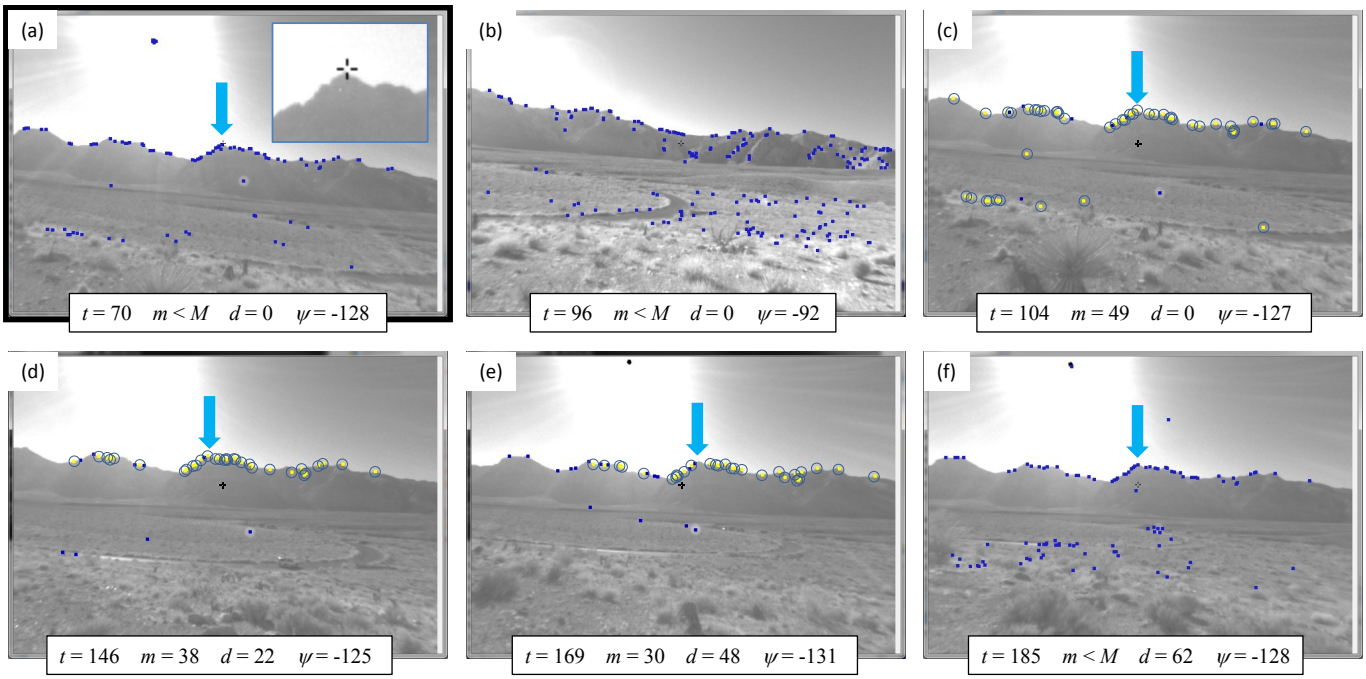

Fig. 6. Illustration of the landmark matching process using imagery from the forward-looking camera. The small cross hair in each image corresponds to the camera's optical axis and is used to align the camera to the known landmark. Image (a) shows the moment when the landmark (the mountain peak shown in the inset and by the large arrow) is clicked, generating a landmark key-frame. In image (b) the user has looked away from the landmark and there are not enough feature matches. In image (c) the user looks back at the landmark and the software automatically re-acquired the landmark by matching $m = 49$ features (shown by circles) to the landmark key-frame. The user then walks 22 m (d) and 48 m (e) from the starting point and the landmark is re-acquired in each case. Image (f) shows that after walking 62 meters the landmark is not re-acquired. Each image is annotated with the elapsed time t in seconds, the number of matched features m, the distance d from the starting point in meters, and the estimated azimuth ψ in degrees. M is the minimum number of matches that are required (in this case, $M = 20$).

measurements, and those sections typically correspond to stationary conditions (i.e., not walking). Overall, when the user requires it, landmark matching can provide a highly accurate orientation measurement.

C. Horizon Matching

The horizon matching (HM) module provides a measurement of absolute orientation by comparing edges detected in the camera imagery with a horizon silhouette edge generated from DTED, using a hierarchical search algorithm initialized at the current orientation estimate from the EKF.

Stein and Medioni [19] demonstrated feasibility of localization from horizon data using fully synthetic experiments. Their method approximated the horizon as a set of line segments and employed several line-fitting tolerances during the matching. In contrast to Stein and Medioni, the method described here uses real-world data and can generate refined orientation measurements at 20 Hz with current hardware. Behringer et al. [20] estimate camera pose by comparing salient points in a horizon silhouette derived from DTED to the visible horizon silhouette extracted from the camera image. The assumption is made that the horizon silhouette shows up as a strong-gradient edge in the image, which is not the case under various lighting conditions and in cases of severe occlusions by foreground objects. The method described here is robust to both of these disturbances because it uses only the more stable parts of the horizon. More recently, Badoud et al. [21] proposed a robust system for pose estimation that finds the best alignment between a 3D terrain model and an input image. An interesting aspect of their system is that they leveraged secondary silhouettes (visible local mountain peaks or ridges that do not coincide with the uppermost visible horizon silhouette) in the terrain data and the image to improve their alignment. They were able to achieve very accurate results over various types of mountainous terrain, but their system is far too computationally expensive, preventing its use in real-time low-SWAP applications.

1) HM Method: The basic principle of the HM method presented here is that given the user's position and a 3D height map of the terrain surrounding him, a corresponding 360-degree horizon can be computed. If accurate alignment can be found between the computed horizon and the horizon extracted from the camera imagery, then the camera's absolute orientation can be determined.

After transforming the DTED into ECEF coordinates, the next step is to determine the corresponding shape of the horizon from the user's estimated current position. This 3D terrain model is then rendered onto a unit sphere centered at the user's position, where the rendering resolution is chosen to match the native resolution of the camera. To support automatic extraction of the horizon silhouette, the 3D terrain model is rendered as a white surface onto a black background, so that

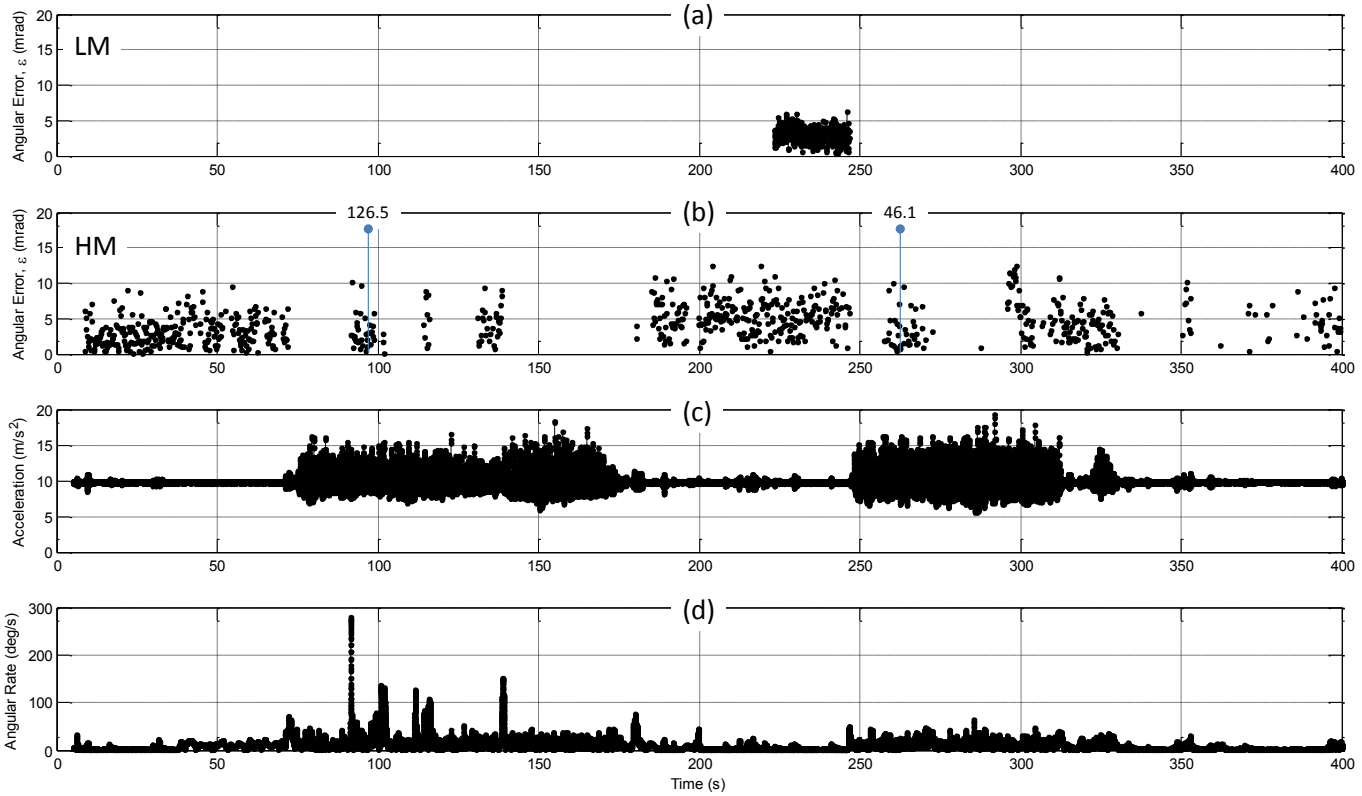

Fig. 7. Sample standalone accuracy performance (as defined in Sec. II-A12) of the LM module (a) and HM module (b) during field tests in the Smoky Mountains. Plots (c) and (d) show the accelerometer and rate gyro data, respectively, to give an indication of dynamic conditions. In this sample, LM has a mean error of 2.9 mrad with a standard deviation of 1.0 mrad. Note that LM is used under stationary conditions (i.e., not walking), when needed by the user. The HM module, on the other hand, functions automatically and produces a measurement as long as there is a horizon available to match. Note, however, the presence of two false positive matches (outliers) in plot (b). In this sample, without including the outliers, HM has a mean error of 4.3 mrad with a standard deviation of 2.6 mrad.

the horizon extraction becomes a simple edge detection. Using the inverse of the camera calibration matrix, each pixel along the horizon is converted to its corresponding image vector, and normalizing these vectors yields a spherical representation of the horizon silhouette.

Given the spherical representation of the horizon silhouette, several optimizations can be performed to improve the computational efficiency. To facilitate data compression and improve processing efficiency, a continuous connected chain is created that represents the 360-degree horizon silhouette. First, edges are extracted from the projected spherical image followed by an edge-following algorithm in the image to define an edge chain. While the edge chain is a very good representation of the horizon, it is also a very dense representation posing computational challenges for the alignment. This leads to a second step in which the pixel-resolution chain is reduced to a much smaller set of line segments that satisify a maximum tangential distance. The resulting piece-wise linear representation typically reduces the complexity of the horizon and greatly boosts the computational efficiency.

To extract a horizon from the camera imagery, edge detection is performed on each undistorted image by first blurring with a Gaussian filter and then using a Sobel filter along both the horizontal and vertical directions. From this, the squared edge response is computed at each pixel location by summing the squares of the vertical and horizontal edge components. Then the image of the squared edge response is blurred again with a Gaussian filter to effectively increase the size of the edges. The last step is to threshold the edge response so that it is equal to one along the edges and zero elsewhere. The threshold is set so that the resulting edges are around five to ten pixels wide (the advantage of which is discussed later). At this point in the process, a pyramidal representation of edge images is also created, which is used later in a course-to-fine search. To create the down-sampled images, a simple bilinear interpolation scheme is applied where the results are then rounded to maintain the binary nature of the edge image. The rationale for this process is that extracting the edges from the imagery is desirable because the actual horizon silhouette is typically an edge within the image. The reason for the thresholding is that in many cases, the strength of the edge along the horizon varies, even within the same frame-to-frame video sequence. The desired approach is to treat a strong edge in the same manner as a weak edge, as each are equally as likely to be the true horizon silhouette.

The next step is to perform an optimization that seeks

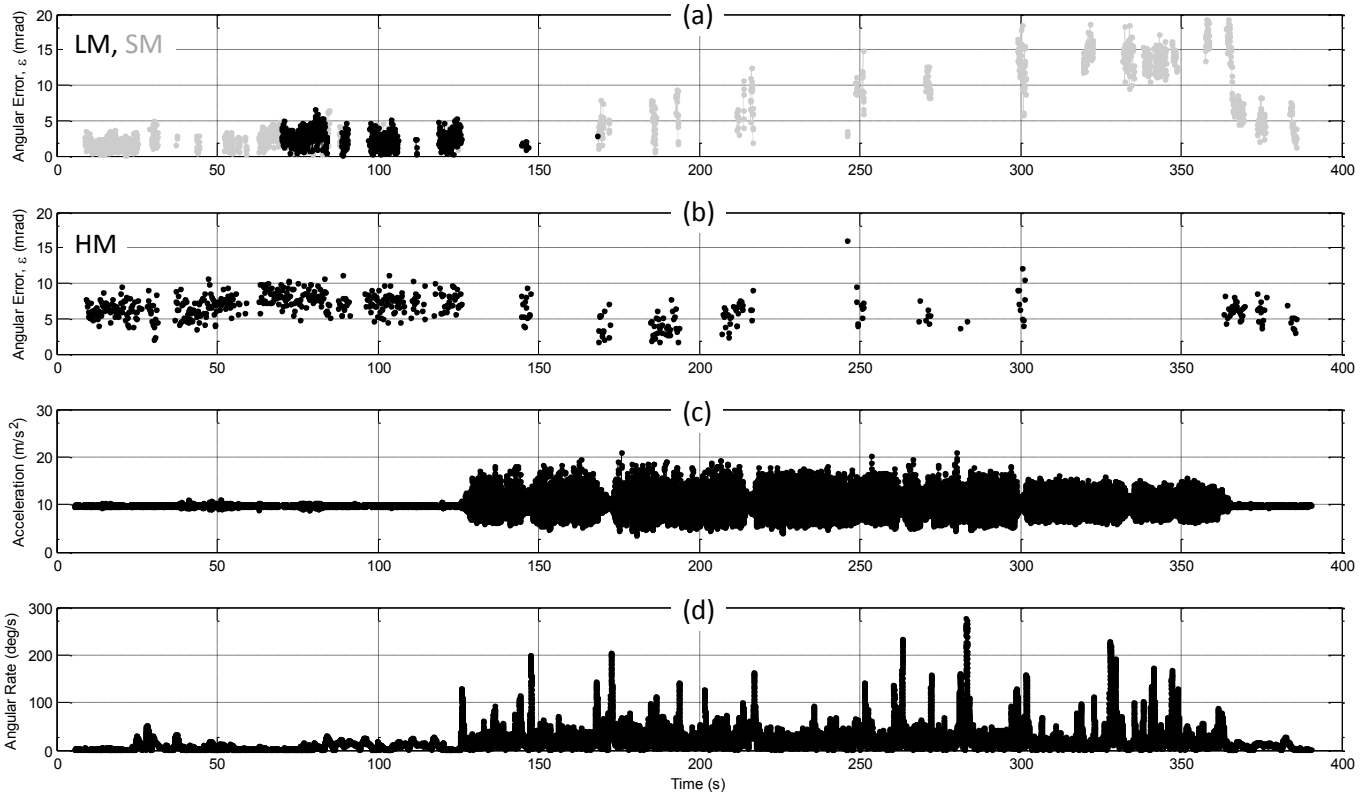

Fig. 8. Sample standalone accuracy performance (as defined in Sec. II-A12) of the LM module (a) and HM module (b) during field tests in the area of Red Rock Canyon, NV. Plots (c) and (d) show the accelerometer and rate gyro data, respectively, to give an indication of dynamic conditions. In this sample, LM has a mean error of 2.5 mrad with a standard deviation of 1.2 mrad. Note that LM is used under stationary conditions (i.e., not walking). It did, however, provide some accurate measurements even after walking some distance (as is also illustrated in Fig. 6), due to the nature of the scene. The HM module functions automatically and produces a measurement as long as there is a horizon available to match. In this sample, HM has a mean error of 6.3 mrad with a standard deviation of 2.0 mrad. Also shown in plot (a) is the standalone accuracy performance of the SM module prototype under development.

the best alignment between the terrain's horizon silhouette and the horizon silhouette from the camera image. This process is initiated using the EKF's current orientation estimate, which is used to transform the horizon silhouette into the expected image. The obtained horizon edge image (given a perfect alignment) would correspond to the observed horizon silhouette in the camera frame. Once the silhouette has been projected onto the image using a specific alignment, a measure of goodness is assigned to this alignment, based on the amount of overlap between the projected horizon and the edges in the camera edge image. This is why wide edges are enforced in the previous image-processing phase. Given that a single-pixel width silhouette is being aligned with the edges in the camera edge image, wide edges are needed to help account for any slight differences between the DTED-based horizon and what is actually seen in the image. For instance, a forest of trees along the top of a mountain ridge will slightly alter the shape of the ridge, but will still exhibit a strong resemblance to the shape of the underlying terrain. The wider edges result in a more robust measure of goodness that allows for slight misalignments without excessive penalty. Additionally, the measure of goodness favors segments of longer overlap as their orientation is more reliable.

To determine optimal alignment, an orientation search is first performed in a region that is centered around the orientation reported by the EKF. To obtain the global maximum in the search region, a hierarchical multi-start gradient ascent technique is used. The search space is first sampled coarsely and uniformly, and several local gradient searches are started from those samples. Once each local search is completed, the maximum of all locals searches is taken to be the global maximum. Then, using a coarse-to-fine approach, the result is up-sampled, and a new search begins at the next highest resolution. When the final search completes, the resulting orientation measurement is produced along with a confidence metric. This metric reflects preference for longer overlapping segments as well as segments that vary in their shape, which is equivalent to a high-gradient entropy of the segment. Before the orientation measurement is sent to the EKF, the corresponding confidence metric has to exceed a relatively high threshold. This is done to prevent measurements coming from false positive matches from corrupting the EKF's measurement update.

2) Standalone HM Results: Standalone results are produced by using the HM orientation measurement, $\boldsymbol{y}_{\mathrm{HM}}$, to compute ϵ, instead of using the EKF's orientation estimate. Horizon

matching typically works best in mountainous environments because of the distinctiveness of the horizon's shape. The HM module was tested in multiple distinctly different mountainous regions including the Smoky Mountains (see Fig. 9) and Red Rock Canyon, NV (see Fig. 6), and was able to tolerate the large differences in horizon shapes. A sample of performance of the HM module in these environments is shown in plots (b) of of Fig. 7 and Fig. 8. HM performed well in both cases, with a mean error less than 7 mrad. Note in Fig. 7 that there are a couple of outliers corresponding to false positives whose confidence metric passed the threshold test. These have the potential of affecting the integrated solution (the effect of the first of the two false positives is obvious as a spike in the error plot of Fig. 10). Though not yet implemented, these isolated outliers could be detected in the EKF by monitoring the statistics of $\boldsymbol{z}_{\mathrm{HM}} = \boldsymbol{y}_{\mathrm{HM}} - \hat{\boldsymbol{y}}_{\mathrm{HM}}$. On the plus side, note in Fig. 9 that the HM module is able to find matches through several occlusions, such as the trees in images (a) and (b), and the parked car in image (e). Overall, when a reasonably distinctive horizon is in view, horizon matching can provide an accurate and robust orientation measurement.

D. Sun Matching

After a recent field test, it was noticed that the Sun appeared in a large portion of the camera imagery as a black spot on an otherwise bright sky (see, for example, image (e) of Fig. 6). In fact, this "eclipsing" phenomenon is characteristic of many CMOS sensors and occurs when the photo-generated charge of a pixel is so large that it impacts the pixel's reset voltage and subsequently the signal-reset difference level presented to the analog-to-digital converter. This results in saturated pixels being incorrectly decoded as dark pixels. Most CMOS sensors include anti-eclipse circuitry to minimize this effect, but this function had been effectively disabled in our camera. The resulting black-Sun artifact prompted an exploration of Sun-based measurements of orientation, and a preliminary SM module was developed that uses the Sun's location in the camera image to generate a measurement of the camera's absolute orientation.

1) SM Method: The basic method consists of the following steps:

1: find pixel coordinates of black-Sun centroid in undistorted camera image
2: convert pixel coordinates into measured Sun vector in b coordinates, $\boldsymbol{s}^b$
3: compute reference Sun vector in n coordinates, $\boldsymbol{s}^n$
4: using EKF's roll estimate as constraint, find C_b^n such that $\mathsf{C}_b^n \boldsymbol{s}^b = \boldsymbol{s}^n$

The camera model shown in Fig. 5 is used in line 2. In line 3, using an astronomical model [22] and knowledge of $\boldsymbol{p}_p$, date, and time, the reference Sun vector is computed as azimuth and zenith angles in the n coordinate system. The Sun-based orientation estimate returned to the EKF is the rotation matrix that aligns the reference Sun vector in n coordinates with the measured Sun vector in b coordinates, as shown in line 4. This requirement only constrains two out of three angular degrees of freedom, so a third constraint is imposed, which is that the roll angle represented in the Sun-based orientation estimate must be the same as the one in the current EKF estimate of orientation. Under this constraint, a gradient-descent optimization method is used to find the rotation matrix C_b^n that most closely satisfies $\mathsf{C}_b^n \boldsymbol{s}^b = \boldsymbol{s}^n$.

2) Standalone SM Results: Standalone results are produced by using the SM orientation measurement to compute ϵ, instead of using the EKF's orientation estimate. Since the Sun matching method is still under development, it is not yet integrated into the system for real-time operation. Therefore, sample results were obtained by post-processing GPS/INS data corresponding to the sample experiment of Fig. 8. The GPS/INS solution in that experiment was used to provide $\boldsymbol{p}_p$ and a roll estimate to the SM algorithm. The resulting output of the SM module is shown in plot (a) of Fig. 8, where the accuracy performance is shown to be similar to that of LM when the user is stationary. As the user walks, performance degrades but is still shown to be better than GPS/INS alone. This degradation may be caused by errors in the GPS/INS estimate of roll being propagated to the SM solution.

Being based on a single data collect and a first-try algorithm, these results are very preliminary. Further development will include exploring different algorithm options (especially regarding other choices of enforced constraint) and testing in a variety of environments and sky conditions. Optimum camera placement and field-of-view strategies must also be evaluated and the robustness of the CMOS sensor's eclipse feature must be determined. At this point, however, it is already clear that Sun-based pose estimates are a promising opportunistic aid to navigation solutions.

III. Integrated System Results

All results presented in this paper are based on real-time pose estimation data collected with the actual system during field tests in unprepared outdoor environments. The corresponding computer hardware consists of an off-the-shelf embedded processing module (SECO QuadMo747-X/T30) with an Nvidia Tegra 3 system-on-chip (SoC) and 2 GB of DDR3L memory on-board. This computing hardware uses about 12 W of power at full load provided by a high-capacity battery, and runs a standard version of Ubuntu Linux 12.04 provided by SECO, which supports "hard-float" binary modules.

Accuracy performance results were shown for the individual vision-aiding measurement modules in Fig. 7 and Fig. 8. In this section, integrated system (i.e., vision-aided) accuracy performance results based on the same two representative data sets is discussed. These results, shown in Fig. 10 and Fig. 11, correspond to system operation of several minutes under a various dynamic and magnetic conditions. Also shown in these figures are indicators of the presence of LM and HM aiding measurements, and the validity of accelerometer (ACC) and magnetometer (MAG) data (based on previously discussed dynamic and magnetic disturbance detection).

Both Fig. 10 and Fig. 11 show the GPS/INS-only (i.e., no vision-aiding) solution performance varying about an offset of

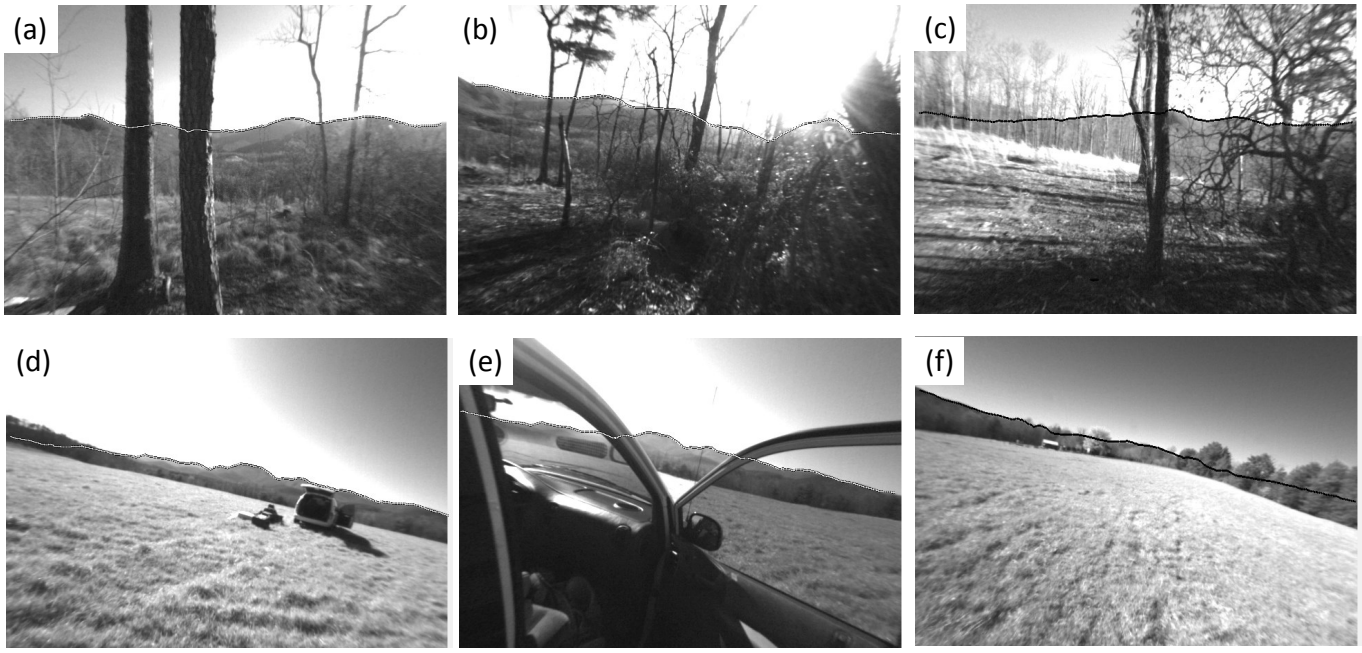

Fig. 9. Examples of HM module matches under different conditions during field tests in the Smoky mountains. In these images, a light horizon line superimposed on the real horizon indicates a successful match (i.e., the confidence metric exceeds a given threshold) and a dark line indicates an unsuccessful one. The threshold is set to a relatively high value to avoid the occurrence of false positive matches, which would corrupt the EKF's measurement update, at the expense of false negatives such as the one in (c). Note that the HM module is able to find matches through several occlusions, such as the trees in images (a) and (b), and the parked car in image (e).

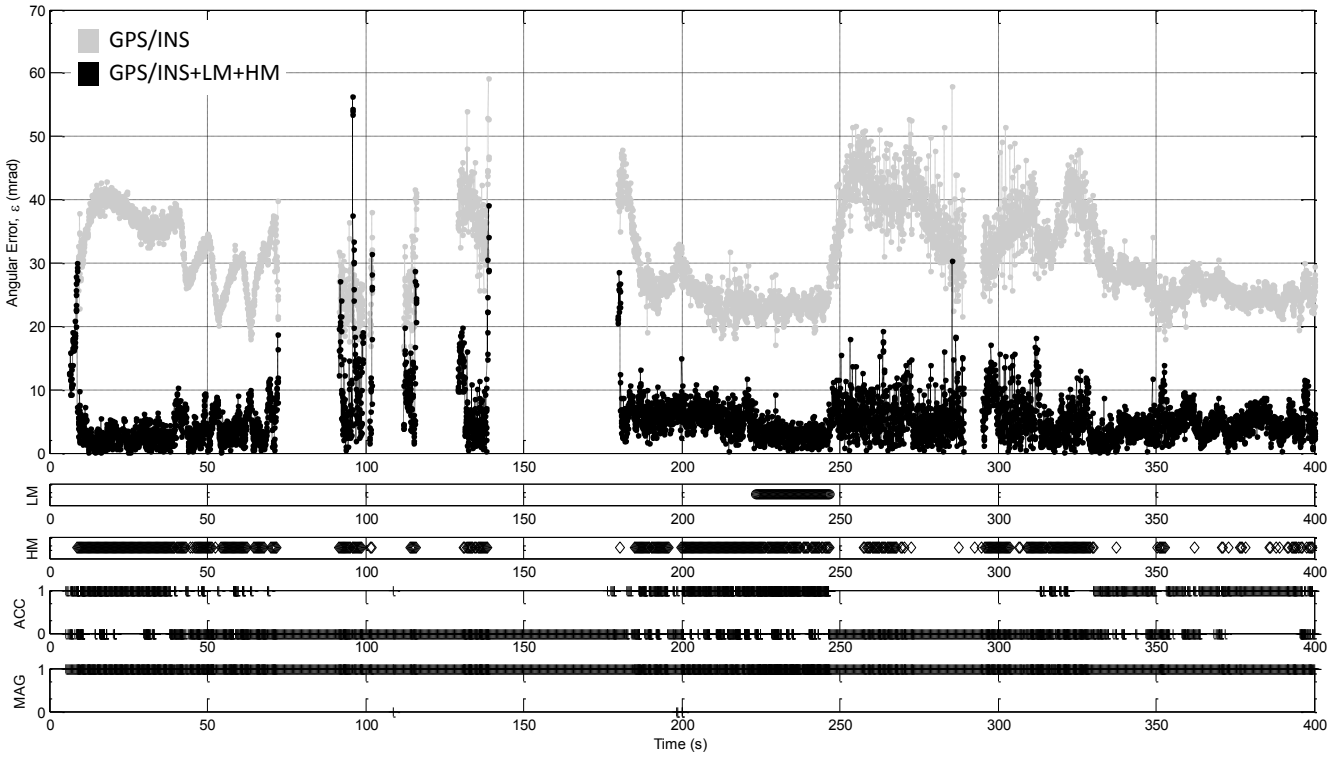

Fig. 10. Integrated system accuracy performance (as defined in Sec. II-A12) corresponding to the data shown in Fig. 7 (Smoky Mountains). Also shown are indicators of the presence of LM and HM measurements, and the validity of accelerometer (ACC) and magnetometer (MAG) data.

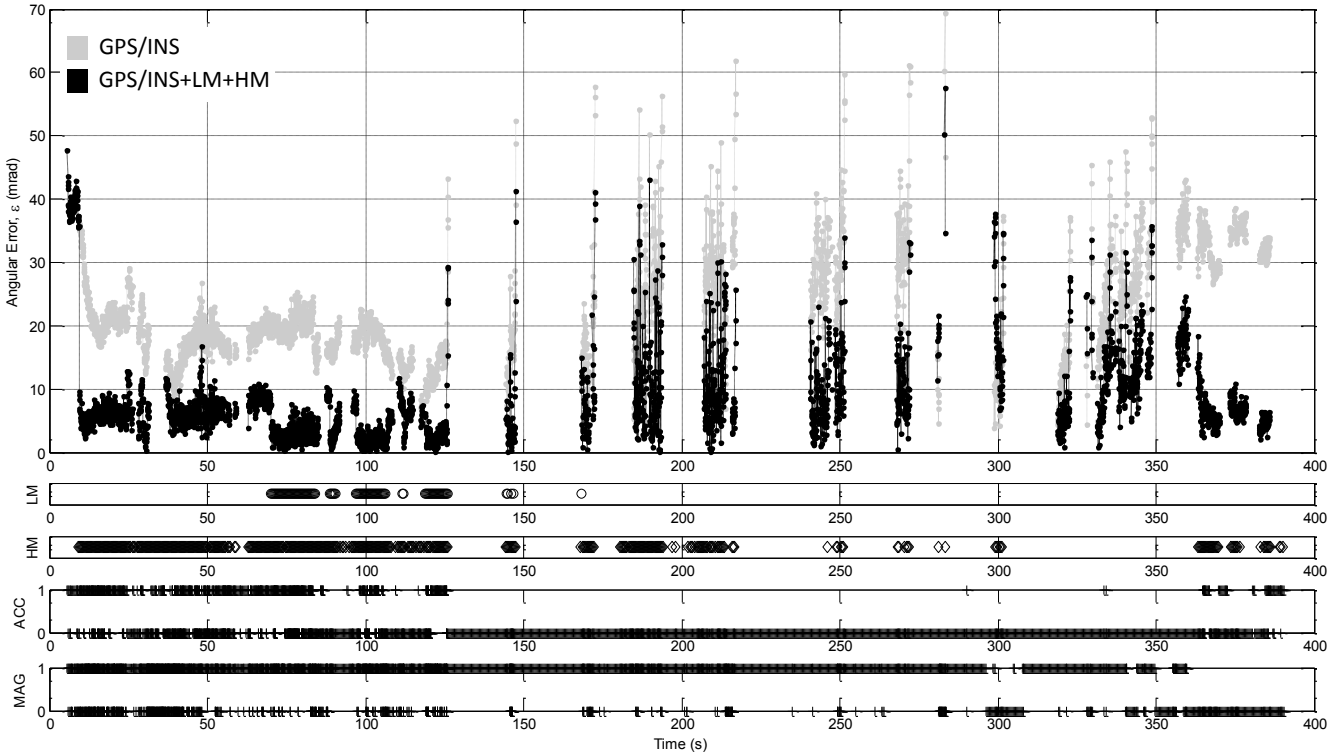

Fig. 11. Integrated system accuracy performance (as defined in Sec. II-A12) corresponding to the data shown in Fig. 8 (Red Rock Canyon). Also shown are indicators of the presence of LM and HM measurements, and the validity of accelerometer (ACC) and magnetometer (MAG) data.

20 mrad to 30 mrad. (Variation about this offset over the experiment are due to varying magnetic and dynamic conditions.) This is representative of our experience in operating in this mode, where systematic errors typically between 10 mrad and 40 mrad have been observed. Since the largest component of this error is in the azimuth direction, its main cause is due to the difficulty of using the Earth's magnetic field as a reference for orientation. This difficulty is two-fold. The first aspect has to do with magnetometer calibration errors. The typical calibration procedure, aimed at correcting hard-iron and soft-iron effects, does not address a residual orientation error that is azimuth-dependent and would have to be corrected by a more burdensome alignment [23], which is impractical in our application. The second aspect has to do with inaccuracies in the assumed model of the Earth's magnetic field, which contribute to the overall error through inaccurate values of magnetic declination and inclination.

The availability of vision-based measurements of absolute orientation provides a way to correct these errors, which is why magnetic bias terms were included in the state vector definition. In fact, Fig. 10 and Fig. 11 show that once the system starts receiving vision-aiding measurements, its pose accuracy increases dramatically. Also, because estimates of magnetic biases are only updated when a vision-based orientation update has occurred, the benefit of vision-aiding measurements persists even when they are no longer available because they have helped to correct the magnetometer measurement model. This is evident, for example, over the ten seconds around $t = 280$ in Fig. 10, where there is a gap in the availability of vision-based measurements, and yet the pose estimate retains its accuracy because of an improved magnetometer measurement model. Without this improvement, the performance would revert back to that of GPS/INS when vision-aiding is not available. (Fig. 11 shows another example at around $t = 240$.)

Rapid error increases visible in both figures are due to sudden head rotations (visible in plots (d) of Fig. 7 and Fig. 8) whose beginning and end cannot be predicted by the forward-prediction process and therefore expose the system's latency as error. (Without forward prediction, these error spikes would be larger.) Note that one of the rapid error increases in Fig. 10, at around $t = 95$, is due to the large HM false positive mentioned previously and shown in plot (b) of Fig. 7. It can also be seen that the best overall performance is achieved when LM is active. This is due to the fact that it is used under stationary conditions and also has a higher intrinsic accuracy than HM. When LM is not active, performance is driven by the accuracy of HM, with areas of higher error variance that correspond to dynamic conditions (i.e., the user walking). In Fig. 11, for example, the user walked continuously over rocky and uneven terrain from around $t = 125$ to around $t = 360$, and a corresponding increase in error variance can be observed due to the difficulty in measuring the gravity vector under those conditions. In the same figure, a number of time periods

with magnetic disturbance can also be observed. Overall, since the performance shown in these figures is representative of what was observed across many other field tests, this vision-aided navigation system consistently showed a significant improvement over the GPS/INS solution, maintaining a mean error below 10 mrad over a wide range of magnetic and/or dynamic conditions.

Exploiting signals of opportunity such as the vision-based measurements described in this paper is a sensible and feasible strategy for greatly expanding the operational envelope of high-accuracy, low-SWAP navigation systems. It is a promising path toward achieving the "Holy Grail" of robust and accurate tracking outdoors, for augmented reality everywhere.

IV. Future Work

Opportunities for future work consist of continuing explorations that were started but left at the preliminary stage, as well as undertaking new investigations based on what was learned during the development of the current system. These include:

- Integration of other available altitude measurements;
- Revisiting tight integration of frame-to-frame vision information;
- Revisiting the possible role and use of vision-based SLAM techniques;
- Development and implementation of enhanced magnetometer calibration;
- Development and implementation of enhanced magnetic measurement error model to provide a means of updating the magnetometer calibration in real-time when absolute orientation measurements are available;
- Continuation of SM investigation and development;
- Continued enhancement of LM and HM algorithms aimed at improving robustness and providing an appropriate measure of uncertainty and/or integrity of the corresponding measurements;
- Development and implementation of enhanced integrity monitoring, to provide timely warning of degraded performance to the user and generate appropriate measures of uncertainty in the state estimate.

Plans for future work also include integrating all of the current software onto the Android platform. Newer processor architectures are also being tested and preliminary results suggest major performance gains.

Acknowledgment

This research was funded by the DARPA ULTRA-Vis program under AFRL contract FA8650-09-C-7909 as well as ARA internal research and development investment. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

[1] G. F. Welch, "History: The use of the kalman filter for human motion tracking in virtual reality," *Presence*, vol. 18, no. 1, pp. 72–91, 2009.

[2] D. Roberts, A. Menozzi, J. Cook, T. Sherrill, S. Snarski, P. Russler, B. Clipp, R. Karl, E. Wenger, M. Bennett *et al.*, "Testing and evaluation of a wearable augmented reality system for natural outdoor environments," in *SPIE Defense, Security, and Sensing*, 2013, pp. 87 350A1–87 350A16.

[3] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 519–535, 2007.

[4] A. I. Mourikis, S. I. Roumeliotis, and J. W. Burdick, "Sc-kf mobile robot localization: A stochastic cloning kalman filter for processing relative-state measurements," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 717–730, 2007.

[5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[6] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, "Vision-based slam: Stereo and monocular approaches," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343–364, 2007.

[7] National Imagery and Mapping Agency, "Nga: Dod world geodetic system 1984: Its definition and relationships with local geodetic systems." [Online]. Available: http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf

[8] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*, 2nd ed., ser. GNSS technology and application series. Boston: Artech House, 2013.

[9] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 561–575, 2007.

[10] T. Ozyagcilar, "Calibrating an ecompass in the presence of hard and soft-iron interference," 2013. [Online]. Available: http://www.freescale.com/files/sensors/doc/app_note/AN4246.pdf

[11] D. H. Titterton, *Strapdown Inertial Navigation Technology*, 2nd ed. AIAA, 2004.

[12] S. Maus, "An ellipsoidal harmonic representation of earth's lithospheric magnetic field to degree and order 720," *Geochemistry, Geophysics, Geosystems*, vol. 11, no. 6, pp. 1–12, 2010.

[13] Gyro and Accelerometer Panel of the IEEE Aerospace and Electronic System s Society, "Ieee std 952-1997, ieee standard specification format guide and test pro cedure for single-axis interferometric fiber optic gyros," 2008.

[14] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Computer Vision–ECCV 2006*, pp. 430–443, 2006.

[15] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "Brief: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2012.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2564–2571.

[17] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, May 1981.

[19] F. Stein and G. Medioni, "Map-based localization using the panoramic horizon," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1992.

[20] R. Behringer, "Improving the registration precision by visual horizon silhouette matching," *Proceedings of the First IEEE Workshop on Augmented Reality*, 1998.

[21] L. Baboud, M. Cadik, E. Eisemann, and H.-P. Seidel, "Automatic photo-to-terrain alignment for the annotation of mountain pictures," *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on DOI - 10.1109/CVPR.2011.5995727*, pp. 41–48, 2011.

[22] I. Reda and A. Andreas, "Solar position algorithm for solar radiation applications," *Solar energy*, vol. 76, no. 5, pp. 577–589, 2004.

[23] J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira, "Geometric approach to strapdown magnetometer calibration in sensor frame," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1293–1306, 2011.