Proceedings of IEEE Virtual Reality '99 (Houston, TX, 13-17 March 1999), 252-259.

A Motion-Stabilized Outdoor Augmented Reality System

Ronald Azuma, Bruce Hoff, Howard Neely III, Ron Sarfaty
HRL Laboratories
3011 Malibu Canyon Rd; Malibu, CA 90265
{azuma, hoff, neely, rsarfaty}@HRL.com

Abstract

Almost all previous Augmented Reality (AR) systems work indoors. Outdoor AR systems offer the potential for new application areas. However, building an outdoor AR system is difficult due to portability constraints, the inability to modify the environment, and the greater range of operating conditions. We demonstrate a hybrid tracker that stabilizes an outdoor AR display with respect to user motion, achieving more accurate registration than previously shown in an outdoor AR system. The hybrid tracker combines rate gyros with a compass and tilt orientation sensor in a near real-time system. Sensor distortions and delays required compensation to achieve good results. The measurements from the two sensors are fused together to compensate for each other's limitations. From static locations with moderate head rotation rates, peak registration errors are ~2 degrees, with typical errors under 1 degree, although errors can become larger over long time periods due to compass drift. Without our stabilization, even small motions make the display nearly unreadable.

1 Motivation

Several prototype Augmented Reality (AR) systems have been built for *indoor* applications such as medical visualization, manufacturing, and entertainment. Representative examples include [5] and [11]; see [2] for more references. In contrast, hardly any *outdoor* AR applications have been explored. If portable, personal AR systems existed, they would open up new classes of applications. For example, a person navigating outdoors (such as a hiker or a solider) must manually read compass and GPS measurements, look at a 2-D map, and mentally match that information with what he sees in the surrounding environment. With a personal outdoor AR system, he could instead see spatially-located icons and labels placed directly over the objects of interest, identifying landmarks, the path to travel, and dangerous areas to avoid, all without increasing his cognitive load. Such personal AR systems could also aid distributed, collaborative teams. Widely-dispersed users lack a common context to share spatially-located information. Radioing other team members to observe the "3rd white building to the right of the church" is useless when they see the town from different vantage points. But personal outdoor AR displays could unambiguously identify the building of interest to each team member. AR may also be a natural interface for some outdoor wearable computer systems, instead of the standard WIMP interfaces that distract the user from the real world rather than complementing it [9].

2 Approach

Outdoor AR applications have rarely been attempted because building an effective outdoor AR system is much more difficult than building an indoor system. First, fewer resources are available outdoors. Computation, sensors and power are limited to what a user can reasonably carry. Second, we have little control over the environment outdoors. In an indoor system, one can carefully control the lighting conditions, select the objects in view, add strategically located fiducials to make the tracking easier, etc. But modifying outdoor locations to that degree is unrealistic, so many existing AR tracking strategies are invalid outdoors. Finally, the range of operating conditions is greater outdoors. The ambient light an outdoor display must operate in ranges from bright sunlight to a moonless night. Environmental ruggedness is vital. Ultimately, we desire the "holy grail" of AR systems: accurate operation indoors, outdoors, and everywhere else.

Figure 1: Virtual labels over outdoor landmarks at Pepperdine University, seen from HRL's Bldg. 250

Our approach is to develop hybrid tracking systems that will move us toward this ultimate goal. No single tracking technology has the performance required to meet the stringent needs of outdoors AR. However,

appropriately combining multiple sensors may lead to a viable solution faster than waiting for any single technology to solve the entire problem. The system described in this paper is our first step in this process. To simplify the problem, we assume the real-world objects are distant (e.g., 500+ meters), which allows the use of differential GPS for position tracking. Then we focus on the largest remaining sources of registration error (misalignments between virtual and real): the dynamic errors caused by lag in the system and distortion in the sensors. Compensating for those errors means *stabilizing* the display against user motion. We do this by a hybrid tracker combining rate gyros with a compass and tilt orientation sensor. This system forms a base from which we will improve, with our collaborators at UNC Chapel Hill, USC, and Raytheon, to strive toward the ultimate goal.

3 Contribution

This system is the first motion-stabilized AR display that works outdoors and achieves tighter registration than any previous outdoor AR system, to our knowledge. Figure 1 shows an example of what our system displays: virtual text labels registered over outdoor landmarks. While our system still has apparent registration errors and limitations on where it can operate outdoors, it still represents a large step forward in outdoor AR tracking.

Our system is most closely related to three sets of previous work. First is Columbia's Touring Machine [6]: an outdoor AR system using a differential GPS and a compass and tilt sensor for tracking. That work focused on potential applications rather than on accurate registration, so the paper does not claim any specific accuracies. Video demos of the Touring Machine show large registration errors (10+ degrees) as the user walks. Our system uses hybrid tracking to stabilize the display, making our registration errors much smaller, and we discuss sensor distortions and calibrations that [6] does not. Second, [1] describes an indoor motion-stabilized display for an optical see-through HMD. While our stabilization strategy is similar, this system differs in the choice of sensors and mathematics required to make this outdoor system work. The different sensors required different calibration and system design decisions. Finally, InterSense builds commercial hybrid trackers. [7] describes an orientation-only sensor that also uses gyros and a compass and tilt sensor. Our work differs in our compensation for sensor distortions, our mathematics, and an actual demonstration and evaluation of registration accuracy in an outdoor AR system. [8] describes an indoor ultrasonic - inertial hybrid tracker but that does not apply to outdoor AR. Concurrent with this work, researchers at the Rockwell Science Center have been building an AR system that achieves registration by recognizing silhouettes at the visual horizon [4].

4 System

4.1 Overview

Figure 2 shows the system dataflow. Three sets of sensors are used: the Omnistar 7000 differential GPS receiver, a Precision Navigation TCM2 compass and tilt sensor, and three Systron Donner GyroChip II QRS14-500-103 rate gyroscopes (±500 degrees per second range). The Omnistar provides outputs at 1 Hz, with 2-3 meters typical error, which we verified against a survey marker in the town of Malibu. The TCM2 updates at 16 Hz and claims ±0.5 degrees of error in yaw. The gyros are analog devices which we sample at 1 kHz using a 16-bit A/D PCMCIA card (National Instruments DAQCard-AI-16XE-50). The other two sensors are read via serial lines. A FutureTech 200 MHz Pentium laptop PC reads the sensors. Section 4.2 describes the sensor distortions and calibration required. The DGPS sensor directly provides the position, but the other two sensor outputs are fused together to determine the orientation, as described in Section 4.3. The user location is then passed to the renderer for display (Section 4.4). The display is a monocular, monochrome optical see-through HMD (Virtual Vision V-Cap) with VGA resolution that we have extensively remounted to provide a rigid relationship between the HMD and the sensors. The entire system renders new images at ~60Hz, matching the update rate of the display hardware. The software is a near real time set of threads and processes running under Windows NT 4.0.

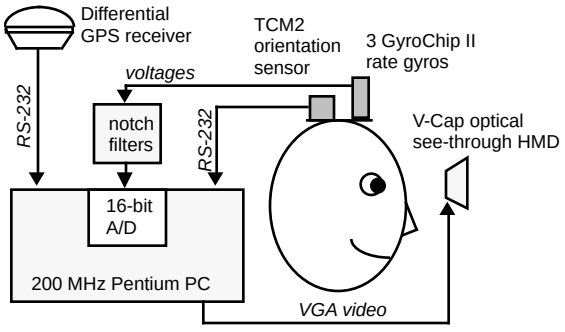

Figure 2: System dataflow

The system operates in both head-worn and hand-held modes. Figure 3 shows the HMD and sensors in a head-worn configuration. However, for ease of demonstrating this system to large groups of people, we also reconfigured the display as a handheld device, with a color video camera (Toshiba IK-M43S) placed where the user's eye normally is, and the video output is shown on a monitor (Figures 4

and 5). We also use the video camera to record the display output, providing the images in this paper.

Figure 3: HMD configuration

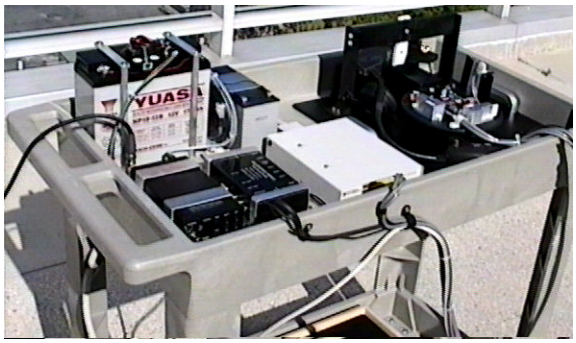

Figure 4: Cart supporting handheld configuration

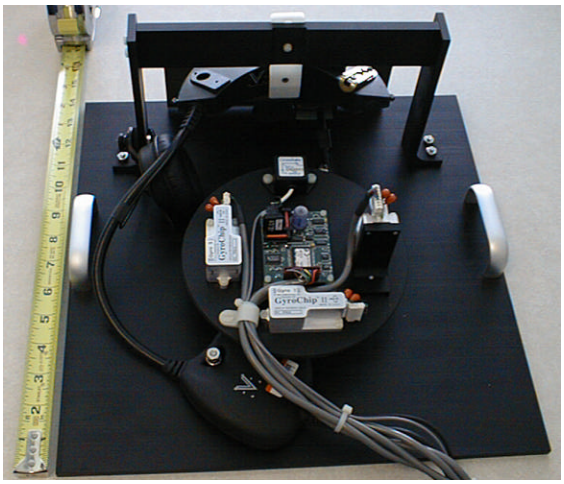

Figure 5: Handheld configuration

4.2 Sensor Calibration

Compass Calibration: We found the TCM2 had significant distortions in the heading output provided by the compass, requiring a substantial calibration effort. Besides the constant magnetic declination, the compass is affected by local distortions of Earth's magnetic field. We built a non-ferrous mechanical turntable to measure these errors at locations we felt were far from any obvious sources of distortion. Figure 6 shows some collected distortion maps, which were taken in pairs at different locations and times. The distortions have peak-to-peak values of about 2-4 degrees. While the patterns within each pair (which were taken within 30 minutes of each other) are similar, they can be quite different across different pairs. Unfortunately, it is difficult to build a working AR display that does not place some sources of magnetic distortion in the general vicinity of the compass. In the real system, compass errors can have peak-to-peak values of 20-30 degrees.

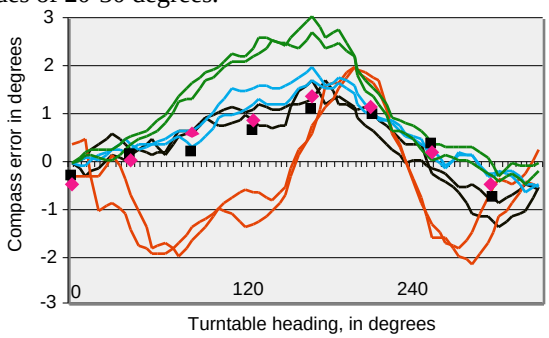

Figure 6: Pairs of compass distortions measured at different locations and times.

Although the distortion pattern is not consistent across all locations and times, the relative consistency between measurements taken 30 minutes apart gave some hope of calibrating the sensor at the beginning of each session. We measure gross distortion maps, like the ones in Figure 6 by sampling the field every 5 degrees (and linearly interpolating). These maps are refined by observing a few known landmarks in the environment. We can compute the true headings to the landmarks by using the known locations of the landmarks and the measured DGPS location of our observation site. These true headings are compared against the measured compass headings. The differences are corrections which are smoothly blended with the original gross distortion map. This provides a correction function mapping compass headings into true headings (on that day, at that site). Although this process requires more manual effort than is desirable, it was necessary to get the best performance we could out of the electronic compass.

Gyroscope Calibration: We measured the bias of each gyroscope by averaging several minutes of output while the gyros were kept still. For scale, we used the specified values in the manufacturer's test sheets for each gyro. The GyroChips have a large internal noise spike around 322 Hz, so we apply active notch filters, designed by Vern Chi

of UNC Chapel Hill, before digitizing the signal [3]. The filters provide over 20 dB of attenuation between 310 and 340 Hz.

To check the bias and scale parameters, we performed an *open-loop* integration of gyroscope output, comparing the integrated values against the rotation measured on a mechanical turntable. After 10-20 seconds of integration, the error was on the order of 0.1 degrees, which gave us confidence in the gyro performance and calibration.

Sensor Latency Calibration: The gyro outputs change quickly in response to user motion, and they are sampled at 1 kHz. In contrast, the TCM2 responds slowly and is read at 16 Hz over a serial line. Therefore, when TCM2 and gyro inputs are read simultaneously, there is some unknown difference in the times of the physical events they each represent.

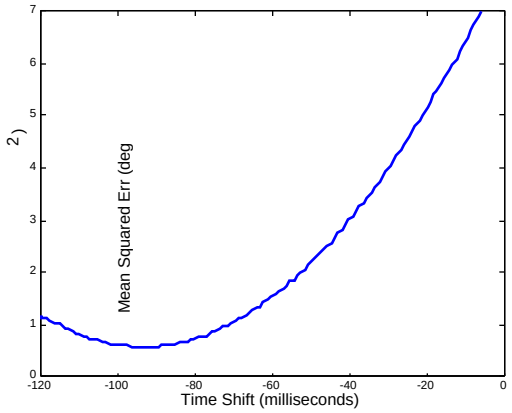

Figure 7: Determining relative latency between TCM2 and gyroscopes

We determined this relative latency in the following way: For several sets of collected TCM2 and gyro data, we integrated the gyro yaw rate to produce heading trajectories. We then integrated the squared difference between the gyro-based heading trajectory and the TCM2-based heading trajectory, varying as a parameter the temporal shift between the two sequences. The result is shown in Figure 7. Across datasets the optimal offset is consistently 92 msec. This latency difference is accounted for in the sensor fusion code.

4.3 Sensor Fusion and Filtering

The goal of sensor fusion is to estimate the angular position and rotation rate of the head from the input of the TCM2 and the three gyroscopes. This position is then extrapolated one frame into the future to estimate the head orientation at the time the image is shown on the see-through display (Figure 8).

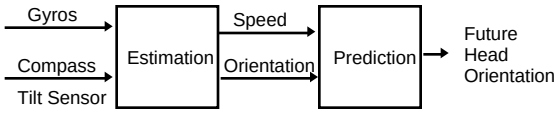

Figure 8: Schematic for fusion and prediction

We relate the kinematic variables of head orientation and speed via a discrete time dynamic system. We define x to be the six dimensional state vector including the three orientation values, as defined for the TCM2 sensor, and the three speed values, as defined for the gyroscopes,

$$x = \begin{bmatrix} r_C & p_C & h_C & r_g & p_g & h_g \end{bmatrix}^T$$

where r, p, and h denote roll, pitch, and heading respectively, and the subscripts c and g denote compass (TCM2) and gyroscope, respectively. The first three values are angles and the last three are angular rates. The system is written,

$$x_{i+1} = A_i\, x_i + w_i\ , \qquad (1)$$

where w_i is noise,

$$A_i = \begin{bmatrix} I_{3x3} & \Delta t\, A_{12}(x_i) \\ 0_{3x3} & I_{3x3} \end{bmatrix},$$

$$A_{12}(x) = \begin{bmatrix} cpc^2 r/a^2 & asrcr\ sp\left(t^2 r + 2/c^2 p\right) & atpc^2 r/c^2 p \\ 0 & a/cp & -atr \\ 0 & atr/cp & a/c^2 p \end{bmatrix},$$

$$a = \frac{1}{\sqrt{1 + t^2 p + t^2 r}},$$

where $c\theta = \cos(\theta)$, $s\theta = \sin(\theta)$, $t\theta = \tan(\theta)$. For example, $cp = \cos(p)$ and $t^2 r = \tan^2(r)$.

r and p are the TCM2 roll and pitch values (r_c and p_c) in x, and Δt is the time step (1 ms). The matrix A_i comes from the definitions of the TCM2 roll, pitch, heading quantities and the configuration of the gyroscopes. A_{12} translates small rotations in the sensor suite's frame to small changes in the TCM2 variables. The derivation is, unfortunately, too long to include here.

To predict the head orientation one frame into the future, we use a linear motion model: We simply add to the current orientation the offset implied by the estimated rotational velocity. This is done by converting the orientation (the first 3 terms of x) to quaternions and using quaternion multiplication to combine them. We will incorporate more sophisticated predictors in the future.

The fusion of the sensor inputs is done by a filter equation shown below. It gives an estimate of x_i every time step (every millisecond), by updating the previous estimate. It combines the model prediction given by (1) with a correction given by the sensor input. The filter equation is,

$$x_{i+1} = A_i x_i + K_i \left(z_{i+1} - A_i \begin{bmatrix} x_{i-92}^{1-3} \\ x_i^{4-6} \end{bmatrix} \right) \quad (2)$$

where K_i is the gain matrix that weights the sensor input correction term and has the form,

$$K_i = K = \begin{bmatrix} g_c I_{3x3} & 0_{3x3} \\ 0_{3x3} & g_g I_{3x3} \end{bmatrix}.$$

g_c and g_g are scalar gains parameterizing the gain matrix. z_{i+1} is the vector of sensor inputs, where the first 3 terms are the calibrated TCM2 measurements (angles) and the last three are the calibrated gyroscope measurements (angular rates). Since the compass input has a 92 msec latency, the first 3 terms of z_{i+1} are compared not against the first three terms of the most recent estimated state (x_i) but against those terms of the estimate which is 92 msec old. During most time steps there is no TCM2 input. In those cases g_c is set to zero, i.e. there is no sensor correction from the compass input.

A word about optimal filtering: Equation (2) above is in the form of a Kalman filter, where K_i would be the Kalman gain, which is based on the relative noisiness of the sensors and model dynamics and provides the weighting which yields the optimal estimation. We can measure the necessary sensor noise covariance matrices, but the process noise is more elusive. Generally, in the Kalman approach the process noise is assumed to be zero mean, Gaussian and white. These assumptions are likely to be invalid, since driving input is lumped in with the noise in (1). Since we did not have a way to accurately determine the process noise, achieving optimality by tuning a Kalman filter did not appear practical. Additionally, calculating the Kalman gain requires several matrix inversions per time step, which is undesirable for a real-time system with limited computing power. Therefore, we chose the alternative approach of using a *constant* gain matrix K, with a small number of parameters, so that empirical tuning of the gain is tractable. The question then is whether exploration of the parameter space can yield a gain matrix which produces desirable filter behavior.

Numerical optimization of the two parameters, g_c and g_g, would be possible if we had "ground truth" against which to compare filter output. Lacking that, we vary the two parameters, first while running the filter in simulation on collected data, and second while using the filter in the actual system. In simulation we use as ground truth the stable compass reading when the system is still. We also can "see" how jittery the TCM2 output is, and look for "smoother" behavior in the filter output. In the integrated system, we can get an informal "feel" of the quality of the performance by watching how closely labels track landmarks in the environment. (In the future we may formalize this using video capture techniques.) Each gain can range from 0 to 1. Small gain values indicate trust in the model over the sensors, and large values indicate trust in the sensors over the model. In practice we found that the gyros were very reliable, and set g_g to 1.0, essentially "integrating" the gyro input. The compass provided a small corrective contribution, preventing drift in the long term. We found it sufficient to set g_c to 0.05. A more complicated filter would adaptively change the gain settings, but at least for this initial system we found that constant gains produced satisfactory results.

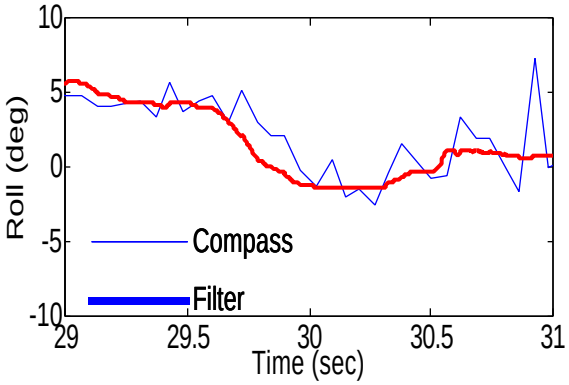

Figure 9: A sequence of roll data. Compass measurements are smoothed by the fusion.

Figures 9, 10, and 11 show graphs of the fusion algorithm's output, with compass measurements for comparison. Note the filter output leads the compass, due to immediacy of the gyro information, versus the compass' 92 msec lag. Gyro input also allows the filter to disregard the noisy tilt sensor values. Figure 11 shows output during settling. Estimation quickly reverts to match compass when gyro outputs are zero. Empirically, the chosen fusion algorithm seems to perform well. Quantification of filter accuracy must wait until "ground truth" is available to compare the filter output against.

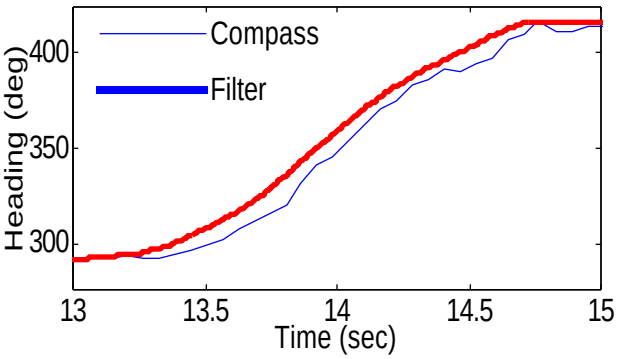

Figure 10: Sequence of heading data. Filter output leads compass measurement.

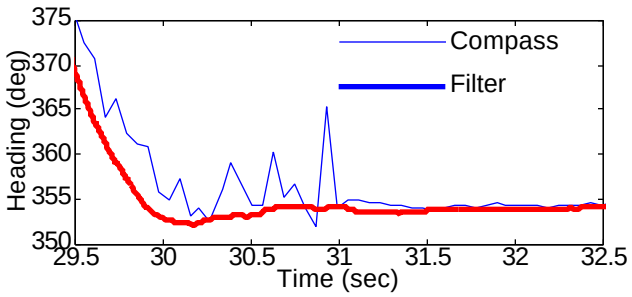

Figure 11: Sequence of heading data. Filter output equals compass when user is still.

This fusion algorithm was influenced by the SCAAT (Single Constraint at a Time) filter [13]. While it is not officially a SCAAT filter, it incorporates individual sensor readings into the filter as they are measured rather than waiting for both types of data to become available. This allows the filter to run at the gyro sampling rate (1 kHz) rather than the compass sampling rate (16 Hz).

4.4 Renderer and Database

The software is primarily organized as one renderer object and one or more database objects. Each database entry stores the Earth-centered (latitude, longitude, altitude) location, Cartesian (x, y, z) location, classification data, and other data for each object. At initialization, an initial user Earth-centered location is established for the current database, from which Cartesian equivalents are created for all current and new database objects, for use in rendering.

The performance design goal for the renderer was to handle a database of up to 50 locations, with up to 10 object labels displayed per frame, while achieving reliable 60 Hz updates. To achieve this goal, all periodic rendering and orientation estimation functions were put within the highest priority thread in the system. In NT4, this is achieved by setting the process class to REALTIME and the thread priority to TIME_CRITICAL. Since: a) this priority is higher than NT4 windows functions, b) our processing completes well within 16 ms, c) thread priorities in REALTIME class processes are not aged, d) we start renderer processing following a voluntary yield of the CPU (thereby assuring a full quantum at the start of execution of the workload) and e) the NT4 Workstation time quantum at maximum boost is 18 ms, we should be assured that the renderer will not be preempted during execution, and will therefore have low periodic variation of execution times and no frame losses. Problems that we encountered with this approach are discussed below.

The renderer uses OpenGL for geometry and DirectDraw 3 for drawing and frame buffer management. Initially, the renderer was a purely OpenGL implementation, but we found that both the Microsoft and SGI OpenGL implementations draw to a back buffer in system memory, then bit block transfer (bit blt) the back buffer to the display. Poor fill rate performance in the laptop display adapter limited display updates to under 10 Hz. Using DirectDraw avoids this problem by drawing to a back buffer in display memory, then flipping the front and back buffers. Fill rate still limited system performance due to back buffer clearing, performed by bit blt from display memory, even though this hardware function was performed in parallel with estimation processing. We kept OpenGL for geometry processing instead of using Direct3D immediate mode because it was far easier to use.

DirectDraw did not, however, solve all problems. One outstanding problem was in implementing execution of the renderer thread at the desired 60 Hz frame rate. We had expected to be able to trigger rendering on an event that would become signaled on completion of display vertical retrace. We found that DirectX does not support this capability, and no such support is currently planned by Microsoft. We next tried to poll for completion of the buffer flip, suspending the thread using the Win32 Sleep() function when the test failed. This was not entirely successful, because the resolution of Sleep() is the clock resolution, which is 10 ms on our system (it can be 15 ms on other systems) [10]. This would force us to limit all rendering and orientation estimation processing to 6 ms to assure 60 Hz operation.

5 Results

We operated this system at four different geographical locations: the patios of two buildings at HRL Laboratories, Malibu Bluffs park, and Webster Field, MD. The Maryland site was for a DARPA Warfighter Visualization demonstration on June 18, 1998. The system proved robust across different geographical locations and for long operation times. At Webster Field, we ran for five continuous hours. Figures 1, 12, 13 and 14 are static images from videotape recorded at these observation points.

For moderate head rotation rates (under ~100 degrees per second) the largest registration errors we usually observed were ~2 degrees, with average errors being much smaller. The biggest problem was the heading output of the compass sensor drifting with time. The output would drift by as much as 5 degrees over a few hours at Webster Field, requiring occasional recalibration to keep the registration errors under control. The magnetic environment at Webster Field was rather harsh (on a runway near many large metal objects and EM sources) so such errors may not be surprising. Overall, however, errors of under one degree were common, placing the virtual labels close

enough to the real objects to unambiguously identify the landmarks to the user.

The stabilization provided by hybrid tracking was a dramatic improvement, and without such compensation the display is unreadable under even small user motions. We ran the demonstration in three different modes: raw compass, calibrated compass, and stabilized. In raw mode, the tracking is based only upon the output of TCM2, with magnetic declination included. The registration errors are over ten degrees in this mode, due to distortions in Earth's magnetic field caused by the local environment and the equipment needed to support the AR display. Then we show the calibrated compass output (running at 16 Hz, which is the limit of the TCM2). With our static calibration routines, the largest registration errors typically observed are ~2 degrees when the display is kept completely still. However, noise in the compass output makes the labels jump around by 0.5 degrees or more, distracting the user. Worse, even small motions make TCM2 output inaccurate, causing large registration errors. These motions can be as small as the vibrations caused when walking around, or even trying to keep one's head still while the wind is blowing. Adding the gyro outputs in our stabilized, hybrid tracking mode mostly overcomes these problems. The display now runs at 60 Hz and the

Figure 12: Three Malibu landmarks observed from HRL Bldg. 254 patio (Lifeguard Station, Jon Douglas Realty Building, Serra Retreat Chapel)

Figure 13: View of Pepperdine from Malibu Bluffs (Hornton Administrative Building, Phillips Tower, Landon Center)

Figure 14: Landmarks from Webster Field airstrip (Control tower and windsock)

virtual labels appear to stay with the real landmarks, even as the user moves around. The smoothness is evident in Figure 9 from Section 4.3, which compares the raw compass input against the fusion module output. Comparing these three modes shows the value of the hybrid tracking approach. Without stabilization, the display is virtually unusable under normal operating conditions.

6 Future Work

While this work is a significant step forward in outdoor tracking, it has many problems and limitations. The system as it currently exists is not easily portable. Much can be done to make the equipment smaller and lighter with lower power requirements. Ultimately, MEMS and custom silicon sensors should provide the required performance in miniature packages. Because the system is bulky, we have only operated it at static locations outdoors (once set up, we do not change positions). As the user walks around, we expect that the changing distortion of Earth's magnetic field will pose a serious challenge. The existing static calibration techniques require too much manual operation and must be broadened to handle changing fields.

Adding additional sensors to our hybrid tracker, especially in the visual domain, should help overcome other problems with the base system. We are not able to walk around all locations because GPS is blocked at places that do not have direct line-of-sight to a sufficient number of GPS satellites. And while the current registration errors

may be small enough for some navigation applications, they are still much larger than is desirable and must be further reduced. Fusion of additional sensor inputs should provide the information needed to overcome these problems. In particular, visual input will be an important component [12] [14]. Better prediction and other compensation for sensor and system delays will also further reduce registration errors.

The display is not bright enough to see in bright sunlight, so we use dark translucent plastic to reduce the ambient light that reaches the display (much like a pair of sunglasses). We anticipate that future displays will be bright enough to handle the contrast outdoors.

Finally, future systems require greater attention to environmental ruggedness and ergonomic issues.

Acknowledgments

This work was mostly funded by DARPA ETO Warfighter Visualization, contract N00019-97-C-2013. Our collaborators on this contract are Gary Bishop, Vern Chi and Greg Welch (UNC Chapel Hill), Ulrich Neumann and Suya You (USC), and Rich Nichols and Jim Cannon (Raytheon). We thank Mike Daily for his support and suggestions, and Vern Chi and Gary Bishop for their notch filter design. We also thank the reviewers for their comments and suggestions.

References

[1] Azuma, Ronald and Gary Bishop. Improving Static and Dynamic Registration in an Optical See-Through HMD. *Proceedings of SIGGRAPH '94* (Orlando, FL, 24-29 July 1994), 197-204.

[2] Azuma, Ronald T. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments 6*, 4 (August 1997), 355-385.

[3] Bainter, James R. Active Filter Has Stable Notch, and Response Can Be Regulated. *Electronics* (Oct. 2, 1975), 115-117.

[4] Behringer, Reinhold. Improving the Registration Precision by Visual Horizon Silhouette Matching. *Proceedings of the First IEEE Workshop on Augmented Reality* (San Francisco, CA, 1 November 1998).

[5] Curtis, Dan, David Mizell, Peter Gruenbaum, and Adam Janin. Several Devils in the Details: Making an AR App Work in the Airplane Factory. *Proceedings of the First IEEE Workshop on Augmented Reality* (San Francisco, CA, 1 November 1998).

[6] Feiner, Steven, Blair MacIntyre, and Tobias Höllerer. A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. *Proceedings of First International Symposium on Wearable Computers* (Cambridge, MA, 13-14 October 1997), 74-81.

[7] Foxlin, Eric. Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter. *Proceedings of VRAIS '96* (Santa Clara, CA, 30 March - 3 April 1996), 185-194.

[8] Foxlin, Eric, Mike Harrington, and George Pfeiffer. Constellation: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications. *Proceedings of SIGGRAPH '98* (Orlando, FL, 19-24 July 1998), 371-378.

[9] Rhodes, Bradley. WIMP Interface Considered Fatal. Position paper at *IEEE VRAIS '98 Workshop on Interfaces for Wearable Computers* (Atlanta, GA, 15 March 1998).

[10] Solomon, David A. *Inside Windows NT, Second Edition*. Microsoft Press, 1998.

[11] State, Andrei, Gentaro Hirota, David T. Chen, Bill Garrett, and Mark Livingston. Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. *Proceedings of SIGGRAPH '96* (New Orleans, LA, 4-9 August 1996), 429-438.

[12] Welch, Gregory. Hybrid Self-Tracker: An Inertial / Optical Hybrid Three-Dimensional Tracking System. UNC Chapel Hill Dept. of Computer Science Technical Report TR95-048 (1995), 21 pages.

[13] Welch, Greg and Gary Bishop. SCAAT: Incremental Tracking with Incomplete Information. *Proceedings of SIGGRAPH '97* (Los Angeles, CA, 3-8 August 1997), 333-344.

[14] You, Suya, Ulrich Neumann, and Ronald Azuma. Hybrid Inertial and Vision Tracking for Augmented Reality Registration. *Proceedings of IEEE VR '99* (Houston, TX, 13-17 March 1999).