Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 864768, 13 pages
http://dx.doi.org/10.1155/2014/864768

Research Article

6DOF Wireless Tracking Wand Using MARG and Vision Sensor Fusion

Harinadha Reddy Chintalapalli, Shashidhar Patil, Sanghun Nam, Sungsoo Park, and Young Ho Chai

Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University, No. 221 Heukseok-Dong, Dongjak-Gu, Seoul 156-756, Republic of Korea

Correspondence should be addressed to Young Ho Chai; yhchai@cau.ac.kr

Received 19 February 2014; Accepted 3 April 2014; Published 16 June 2014

Academic Editor: Sabah Mohammed

We present a low cost battery-powered 6-degree-of-freedom wireless wand for 3D modeling in free space by tri-axis Magnetic, Angular Rate, Gravity (MARG) and vision sensor fusion. Our approach has two stages of sensor fusion, each with different algorithms for finding 3D orientation and position. The first stage fusion algorithm, a complementary filter, utilizes MARG sensors to compute 3D orientation relative to the direction of gravity and earth's magnetic field in a quaternion format, which was adjusted with compensations for magnetic distortion. The second stage fusion algorithm, a Kalman filter, utilizes accelerometer data and IR marker velocity to compute 3D position. In order to compute the IR marker linear velocity along the optical axis (the z-axis), we present a simple and efficient image-based technique to find the distance of the object from the camera using blob area pixels in the image. Our fusion (inside-in and outside-in) approach efficiently solves short time occlusion, needs of frequent calibration, and unbounded drift problems involved in numerical integration of inertial sensors data and improves the degrees of freedom at low cost without compromising accuracy. The results are compared with a leading commercial magnetic motion tracking system to demonstrate the performance of the wand.

1. Introduction

There has been increasing research over the last decade in using 6DOF motion tracking devices for 3D spatial sketching and modeling [1–4] in immersive virtual reality (VR) environments. Other efforts include 2D tablet screens used to draw a 2D sketch that is processed into 3D designs [5, 6]. The objective of practical spatial drawing and editing in 3D demands low-cost, precision, small size, and ease of use. Existing professional motion tracking systems that use electromagnetic, ultrasonic, optical, inertial, and multiple-sensor technologies [7] are too expensive for commercial 3D immersive VR and modeling and require a degree of technical knowledge to use them. Outside-in stereo vision has been widely used for 3D modeling, but this system often suffers from occlusion and interference and apparent loss of DOF. Any accidental change in the position of a camera after calibration requires complete recalibration [8].

3D motion-based human computer interaction (HCI) has long been an active research topic in VR, and it has been shown that 3D interfaces can be useful in many consumer-level applications such as home gaming [9] and 3D user input [10–12]. Emerging demands for rich interaction have led to the development of handheld pointing motion interface devices [13, 14]. These commercial devices incorporate micro-electro-mechanical system (MEMS) inertial sensors such as accelerometers and gyroscopes, and their contributions are limited to gesture recognition, rotation, and vision sensing as for 3D position. These devices are aimed to interact with 3D digital media content and motion gaming and are unsuitable for 3D modeling and editing in free space, which requires precise 6DOF motion sensing.

The main technological bottleneck that limits accuracy in computing position and orientation from MEMS inertial sensors is the drift caused by numerical integration of acceleration and angular rate [15–17]. However, inertial sensors are well known for their short term precision, high-frequency data rates, and size. To leverage these advantages, benefits of sensor fusion techniques using additional sensors have been proposed by researchers in the areas of navigation [18, 19] and motion capture [20].

Another recent innovative work, called MEMSEye, uses MEMS-mirror-based optical 3D tracking [21]. Combination of two or more MEMSEye units can track light sources such as IR light-emitting diodes (LED) and corner cube retroreflectors (CCRs). By triangulating the tracked object's relative position from each unit, its 3D position can be computed in relatively large volumes with submillimeter precision at update rates of >20 kHz. However, a fully functional unit using this technology platform costs more than one thousand dollars, making it unaffordable for a wide range of users.

With increasing interest in 3D display devices, a simple and low-cost solution that can provide enough precision and flexibility is not yet available for 3D modeling. Our work aims to advance 6DOF motion sensing by two-stage multisensor (MARG and vision) fusion to make use of their complementary properties, which was inspired by multisensor navigation systems. So it is essential to acquire accurate timestamp information to synchronize MARG measurements and the camera. In order to do that, we use a hardware triggerable IR camera.

This paper is organized as follows. Section 2 gives the description of the proposed system design and the details of the wireless wand architecture. Section 3 explains first stage sensor fusion for 3D orientation using MARG sensors and the magnetic distortion correction technique. In order to find the linear velocity of the IR marker along the camera z-axis, we explain our image-based technique in Section 4, followed by a description of the second stage sensor fusion, and performance comparison results with a leading commercial motion tracking system in Sections 5 and 6, respectively. Section 7 concludes our approach. Throughout the paper, a notation system of leading superscripts and subscripts similar to [22] is used to denote the relative frame of orientations and vectors. For example, ${}^{e}_{b}q_{t}$ describes the quaternion of sensor body frame b relative to the earth frame e at time t, and ${}^{e}A_{\text{est}}$ is an estimated vector described in frame e. The $\otimes$ operator denotes a quaternion product.

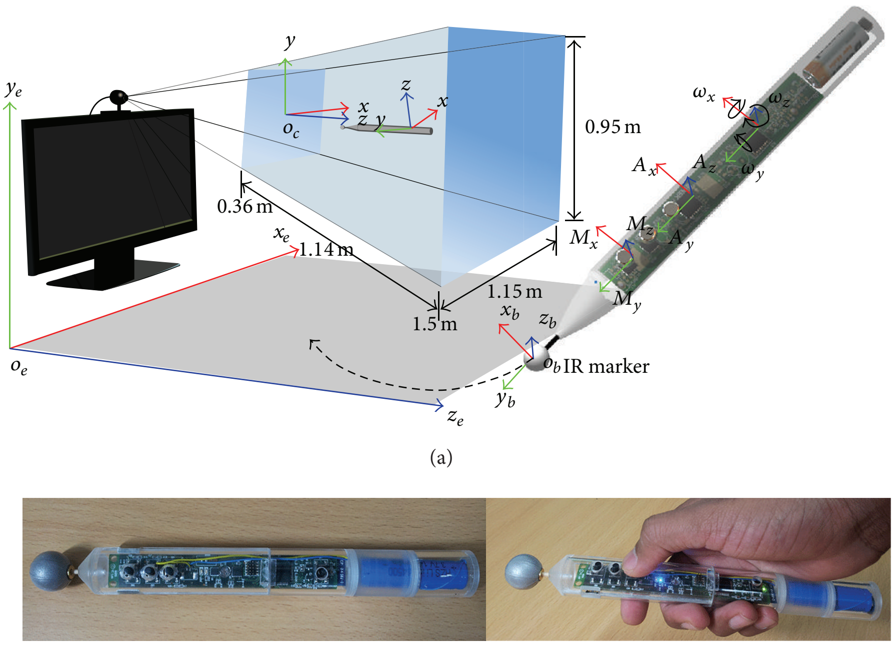
2. System Design

2.1. System Overview. The wireless wand is designed to be used in front of a computer monitor with a camera attached on top of it as shown in Figure 1(a). We use a wide field of view camera which has an IR filter and 640 × 480 resolution to capture the spherical IR marker located at the tip of the wand at 50, 75, and 100 frames per second (FPS). The wand is equipped with triaxis MARG sensors, temperature sensor, and a microcontroller communicates with them to gather a full set of sensor data, from which it computes 3D orientation that is sent in data packets to the computer via wireless Bluetooth link. Data packets contain the first stage sensor fusion output, 3D orientation in quaternion format, raw and calibrated individual sensor data, and buttons and LEDs status information as shown in Figure 2(a). A server program running in the computer communicates with both the wand and the camera as shown in Figure 2(b). In server program, an image-based blob tracking module uses thresholding technique for camera images to find the location and area of the IR marker in image in terms of pixels. The data preprocessing module uses our image-based technique which is explained in Section 4 for finding IR marker velocity along the optical axis (z-axis) of the camera using the area of the marker from the blob tracking result. Also, a technique to calculate gravity vector from the quaternion is incorporated into preprocessing module as MEMS triaxis accelerometer senses gravity plus translational acceleration of wand. The sensor fusion algorithm in Figure 2(b) has a 9-state Kalman Filter (KF) algorithm to compute 3D position. The inputs to the KF are triaxis translational acceleration and velocity vectors from preprocessing module. The server program (API updater) implemented in C++ serves 3D position, orientation, and buttons and LEDs statuses to any application for 3D interaction and modeling. The following section presents the architecture of the wand in detail.

2.2. Wand Architecture. The wand shown in Figure 1(b) has a triaxis digital 16-bit gyroscope, 12-bit accelerometer, and 12-bit magnetometer, each with its own respective selectable ranges of up to ±2000°/s, ±8 g, and ±8.1 G. Also, a digital 16-bit thermometer is incorporated into the wand for adaptive compensation of time varying temperature biases in the MEMS sensors. All the sensors, buttons, status LEDs, and Bluetooth transceivers are connected to a microcontroller for collecting, controlling, and processing data as shown in Figure 3. Wand's firmware incorporates MARG sensors calibration routines and data for computing 3D orientation in order to compensate for sensors biases.

3. MARG Sensor Fusion for 3D Orientation

To compute drift-free measurement of 3D orientation relative to the direction of gravity and earth's magnetic field, researchers proposed several algorithms using MARG sensors [23–26], also known as an attitude heading reference system (AHRS). A complementary filter using low-cost MEMS inertial measurement unit (IMU) with magnetometer was proposed [23] with deep mathematical basis to compute 3D orientation in a direction cosine matrix (DCM) and quaternion form [24]. Though this algorithm showed how a magnetometer can be used along with an IMU (gyroscope and accelerometer) to compute 3D orientation relative to the earth direction of gravity and magnetic field, it was not able to correct drift as was intended due to lack of a compensation technique for magnetic distortions resulting from nearby sources such as metal structures or power supply buses. Several investigations [25, 27] have shown that substantial errors may be introduced by magnetic distortions in orientation estimated from MARG sensors. By adapting a technique proposed in [22] for compensating for magnetic distortions (termed soft iron errors), a complementary filter



(b)

FIGURE 1: System setup to work with wireless wand; (a) working volume of wand in front of camera and different frame of references involved; (b) wand.

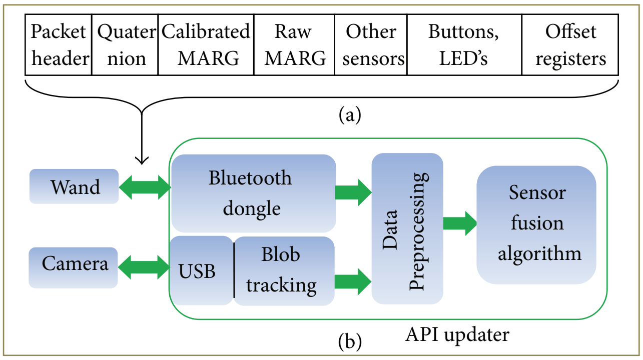


FIGURE 2: Data packet and API: (a) data packet format sent from wand; (b) API to communicate and process data from the wand and camera.

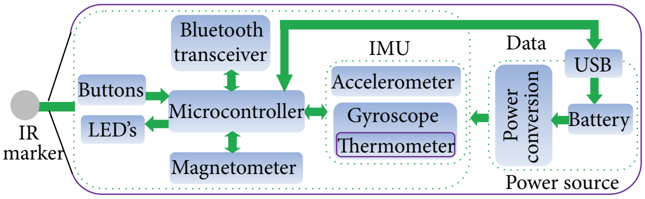


FIGURE 3: Wand architecture.

algorithm in quaternion form has been implemented on a low-power hardware board in the wand system.

The potential advantage of this algorithm is in correcting drift in orientation computed from gyroscope measurements using an additional reference orientation computed from the accelerometer and magnetometer, by successfully incorporating magnetic distortion compensation without any singularity problems. One more key advantage of this technique is that it eliminates the need for the direction of earth's magnetic field to be predefined, which has been a potential disadvantage of other algorithms [25, 26]. The block diagram shown in Figure 4 represents the first stage sensor fusion AHRS algorithm, in which red color box indicates the magnetic distortion compensation technique. More details of the algorithm are given in the Appendix. The compensation technique is described as follows:

$$ {}^{b}M_t = \begin{bmatrix} 0 & m_x & m_y & m_z \end{bmatrix}, \qquad {}^{b}_{e}q_t = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}, \quad (1) $$

$$ {}^{e}H_t = \begin{bmatrix} 0 & h_x & h_y & h_z \end{bmatrix} = {}^{b}_{e}q_{t-1} \otimes {}^{b}M_t \otimes {}^{b}_{e}q^{*}_{t-1}, \quad (2) $$

$$ {}^{e}B_t = \begin{bmatrix} 0 & \sqrt{h_x^2 + h_y^2} & 0 & h_z \end{bmatrix}, \quad (3) $$

where ${}^{b}M_t$ is the normalized magnetic field vector, computed from the output of magnetometer of the wand body frame at time t. ${}^{b}_{e}q_{t-1}$ and ${}^{b}_{e}q^{*}_{t-1}$ are the normalized quaternion output of the algorithm and its conjugate at previous time step $t-1$, respectively. In (2), the direction of earth's magnetic field ${}^{e}H_t$ is computed, which may represent erroneous inclination and can be corrected if algorithm's reference direction of earth's

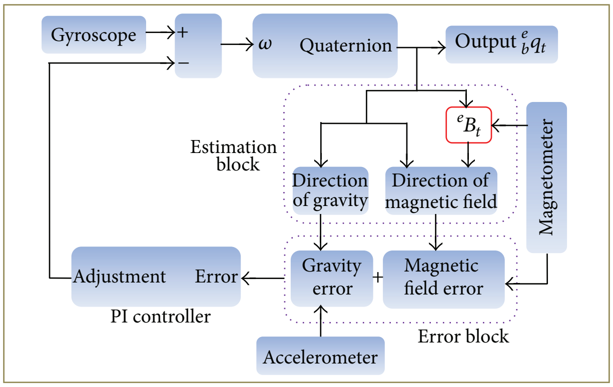


Figure 4: Complementary filter block diagram.

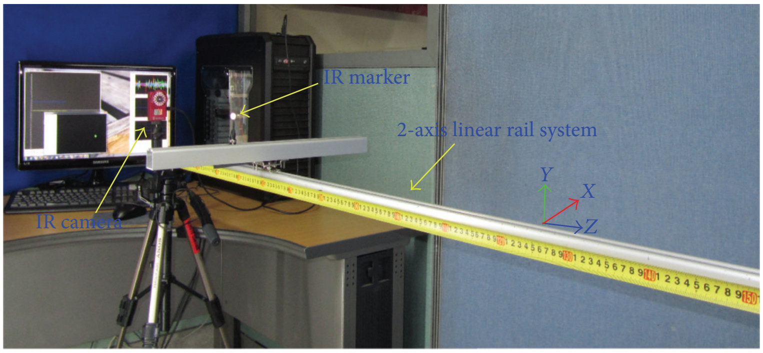


Figure 5: Experimental setup to observe object height and area.

magnetic field $^{e}B_t$ is of the same inclination. Equation (3) ensures that any magnetic disturbances are limited to only affect the estimated heading component of orientation.

4. IR Marker Velocity Tracking

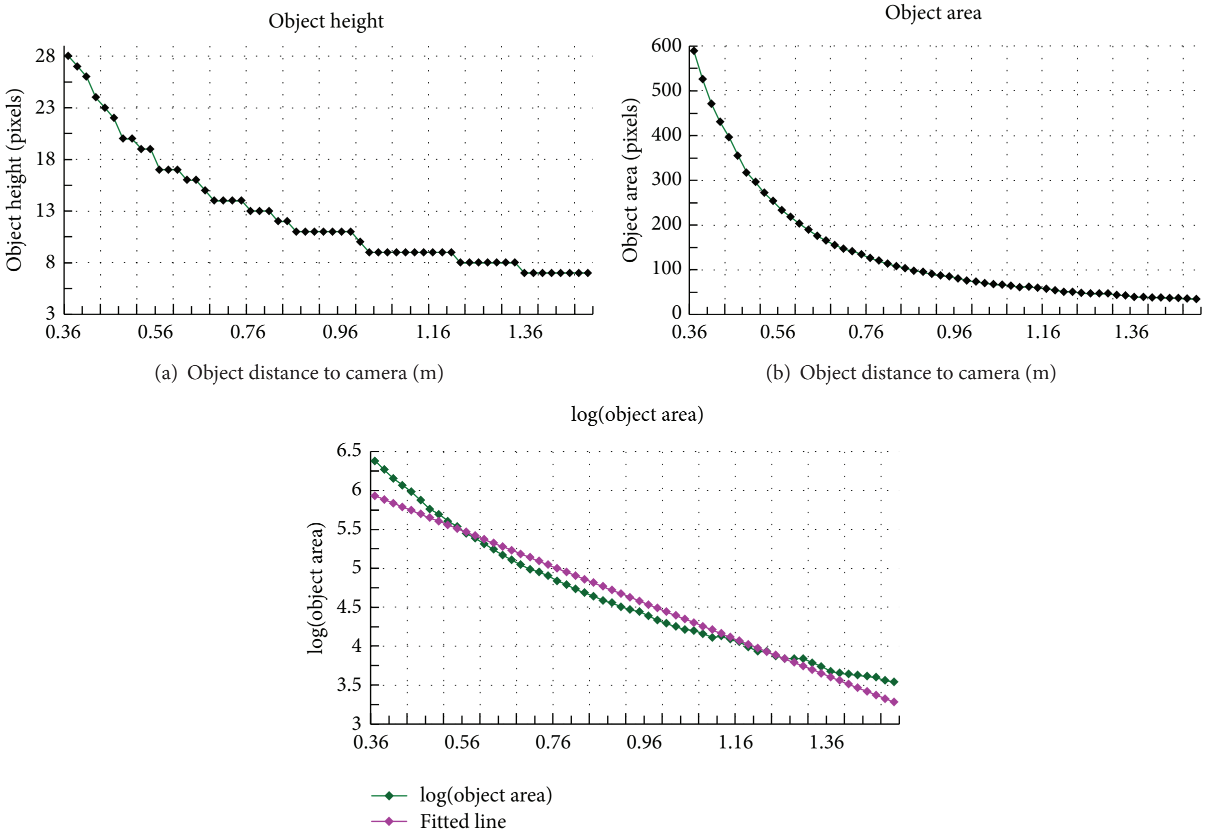
In order to compensate for the numerical drift that results from double integration of acceleration measurements of triaxis accelerometer, we compute 3D velocity of the IR marker from the output of a blob tracking module, which provides the width, height, area, and 2D position $({}^{c}p_x, {}^{c}p_y)$ of the IR marker in pixels after calibrating the image for distortion correction. Now, finding velocities ${}^{c}v_x$ and ${}^{c}v_y$ using position is straightforward in x- and y-axis directions of the camera frame of reference O_c, as follows:

$$ {}^{c}v_x = \frac{{}^{c}p_{x|t-1} - {}^{c}p_{x|t}}{\Delta t}, \qquad {}^{c}v_y = \frac{{}^{c}p_{y|t-1} - {}^{c}p_{y|t}}{\Delta t}, \tag{4} $$

where $({}^{c}p_{x|t-1}, {}^{c}p_{y|t-1})$ and $({}^{c}p_{x|t}, {}^{c}p_{y|t})$ are the 2D positions in camera frame of reference at the previous and present time step, respectively, and $\Delta t = 1/\text{FPS}$. There are two methods to find object position in the z-axis using images: stereovision and monovision. We propose a simple and efficient experimental method to find object distance using an object's area in a single image, and using this distance we find the velocity of a marker along z-axis.

Finding the distance of a specific-shaped object using a single image has been proposed in [28, 29]. Object's height in a thresholded binary image has been used to determine its distance from the camera using rectangular, triangular, cylindrical, and spherical shaped objects. However, object's size was relatively large, with minimum diameter of 0.65 m for a spherical object. For our experiment, we used an industry-standard IR camera with uniform radiation capability to illuminate a retroreflective spherical marker with 0.01 m diameter. Figure 5 shows the experimental setup used to determine how the IR marker object height and area pixels in the image change with varying distance from the optical center of the camera in the z- and x- axes. Initially, the marker is positioned exactly at the optical center of the camera and is moved away from the camera on a 2-axis linear rail system.

It is found that the object height and area decrease exponentially with increasing distance (0.36 m to 1.5 m) from the camera along the z-axis, as shown in Figures 6(a) and 6(b), respectively. But, from Figure 6(a), it is clear that the object height does not change continuously with increasing distance when compared to the area of the object; this



(a) Object distance to camera (m)

(b) Object distance to camera (m)

(c) Object distance to camera (m)

Figure 6: Object height and area with increasing distance from camera.

is the main reason for not using methods presented in other previous schemes [29]. It is also observed that the area measurements are repeatable at any particular intensity, exposure, and threshold settings of the camera. However, in order to calculate linear velocity, it is essential to linearize

$$d = -0.0232577929 \log(a) + 6.7653875804, \tag{5}$$

$$^{c}v_z = \frac{(d_{t-1} - d_t)\,k}{\Delta t}. \tag{6}$$

Figure 6(b) is done by taking the logarithm. Analysis of Figure 6(c) leads to the fact that the object depth has a direct relationship with its logarithmic pixel area. Now, we can find the best fitting linear polynomial by linear regression to find the object distance to the camera, which is given in (5), where d is object distance and a is the object area in pixels. Having object distance, we can find the velocity along z-axis using (6), where k is a constant (scaling factor) determined by observation to get velocity in m/s units.

However, at a particular distance from the camera, when the marker moves laterally to the camera (perpendicular to optical axis) from camera center in either x- or y-axis direction, the object shape in image loses circularity, and the object area is not constant; it varies as shown in Figure 7. This significantly affects the velocity $^{c}v_z$ even if the marker moves only in the x- and y-axis. For this, instead of finding an analytical method based on blob tracking results to solve this problem, we exploit sensor fusion to compensate for the error introduced in the measured z-axis velocity.

The inertial navigation module shown in Figure 10 computes position and velocity estimates using translational accelerations and corrections obtained from a KF algorithm. The estimated z-axis velocity from this module, v_{zest}, can be taken as a corrective reference for two reasons: estimated velocity does not have significant integral drift because correction from the KF is used to compute it; and next, the frame rate of camera is less than the update rate of the sensor fusion algorithm (equal to the sampling rate of inertial sensors). Based on v_{zest}, lower and upper thresholds t_l and t_u, a discriminating window is applied for $^{c}v_z$ to compute valid z-axis velocity $^{c}v_z'$ using (7). Where thresholds t_l and t_u are determined by

$$^{c}v_z' = \begin{cases} 0 & \text{if } v_{\text{zest}} > t_l, v_{\text{zest}} < t_u, \\ \dfrac{v_{\text{zest}} + {}^{c}v_z}{2} & \text{otherwise,} \end{cases} \tag{7}$$

$$^{c}V = \begin{bmatrix} ^{c}v_x & ^{c}v_y & ^{c}v_z' \end{bmatrix} \tag{8}$$

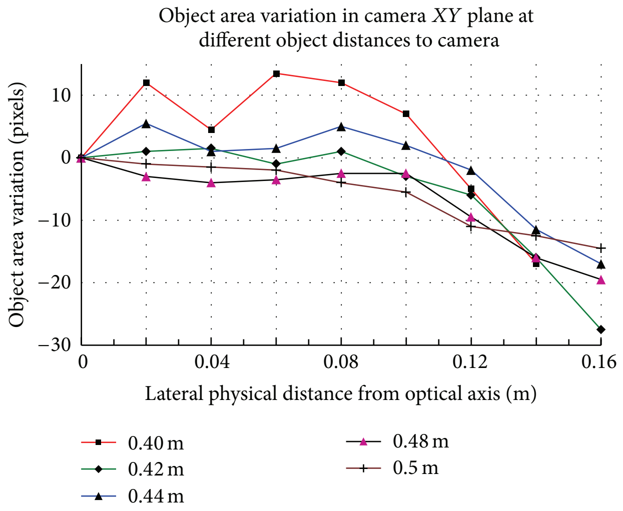


Figure 7: Object area at different depth when object moved laterally to camera from its center.

observing v_{zest}, which implies that below these thresholds v_{zest} has only integral drift resulting from noise in the translational acceleration. Figure 8 shows the velocity graphs for the marker moving back and forth in the x, y, and z directions, before and after applying (7). If v_{zest} falls below both thresholds, ${}^{c}v_z$ is treated as error and is nullified; otherwise we take the mean of both, which is better than trusting noisy ${}^{c}v_z$. Figures 8(a) and 8(b) show the affected ${}^{c}v_z$ and estimated velocity v_{zest} when the marker is moved back and forth along the x- and y-axes, and their corrected counterparts in Figures 8(d) and 8(e). Similarly, Figure 8(c) shows velocities when the marker is moved along z-axis and its corrected counterpart in Figure 8(f), which shows effectiveness of (7) in reducing noise.

5. Sensor Fusion for 3D Position

The MEMS triaxis accelerometer measures acceleration of the wand in the body (moving) frame of reference O_b and has two components, translational acceleration (actual acceleration), and gravity, which is a function of the 3D orientation of the sensor. There are different ways to remove gravity from acceleration [14, 30]. In this section we first explain gravity removal from acceleration which is a part of the data preprocessing module of Figure 2. The next step is a KF sensor fusion algorithm to find 3D position from the translational acceleration vector and velocity vector ${}^{c}V$.

5.1. Gravity Removal in Acceleration. A conditional offset filter [14] to remove gravity from acceleration may be sufficient when its output is used only for gesture recognition, but is not optimal in terms of responsiveness (depends on past data) and accuracy (gravity still persists in transition regions). This affects accuracy when our interest is precise position computation from translational acceleration. The following method, using a quaternion ${}^{e}_{b}q_t$, computes gravity according to (9) and (10). Now, using (11), we can remove gravity from the acceleration vector ${}^{b}A$ to find the translational acceleration vector ${}^{b}A_r$ of moving O_b. Figure 9(a) shows the acceleration of the wand in a stationary state A and arbitrary complex rotation regions B and C. Figure 9(b) indicates how close the computed gravity is to the accelerations. The result after gravity removal in the respective regions is shown in Figure 9(c), where region A contains only noise, and other parts with magnitude greater than region A contain translational accelerations occurring during rotation of wand. Now, body frame translational accelerations have to be transformed to O_e according to strap-down kinematics theory using (12). Equation (13) converts ${}^{e}A$ to units of m/s^2.

Consider the following:

$$ {}^{b}_{e}q_t^{-1} = \left[q_1' \;\; q_2' \;\; q_3' \;\; q_4'\right] = \frac{{}^{e}_{b}q_t^{*}}{\left|{}^{e}_{b}q_t\right|^{2}}, \tag{9} $$

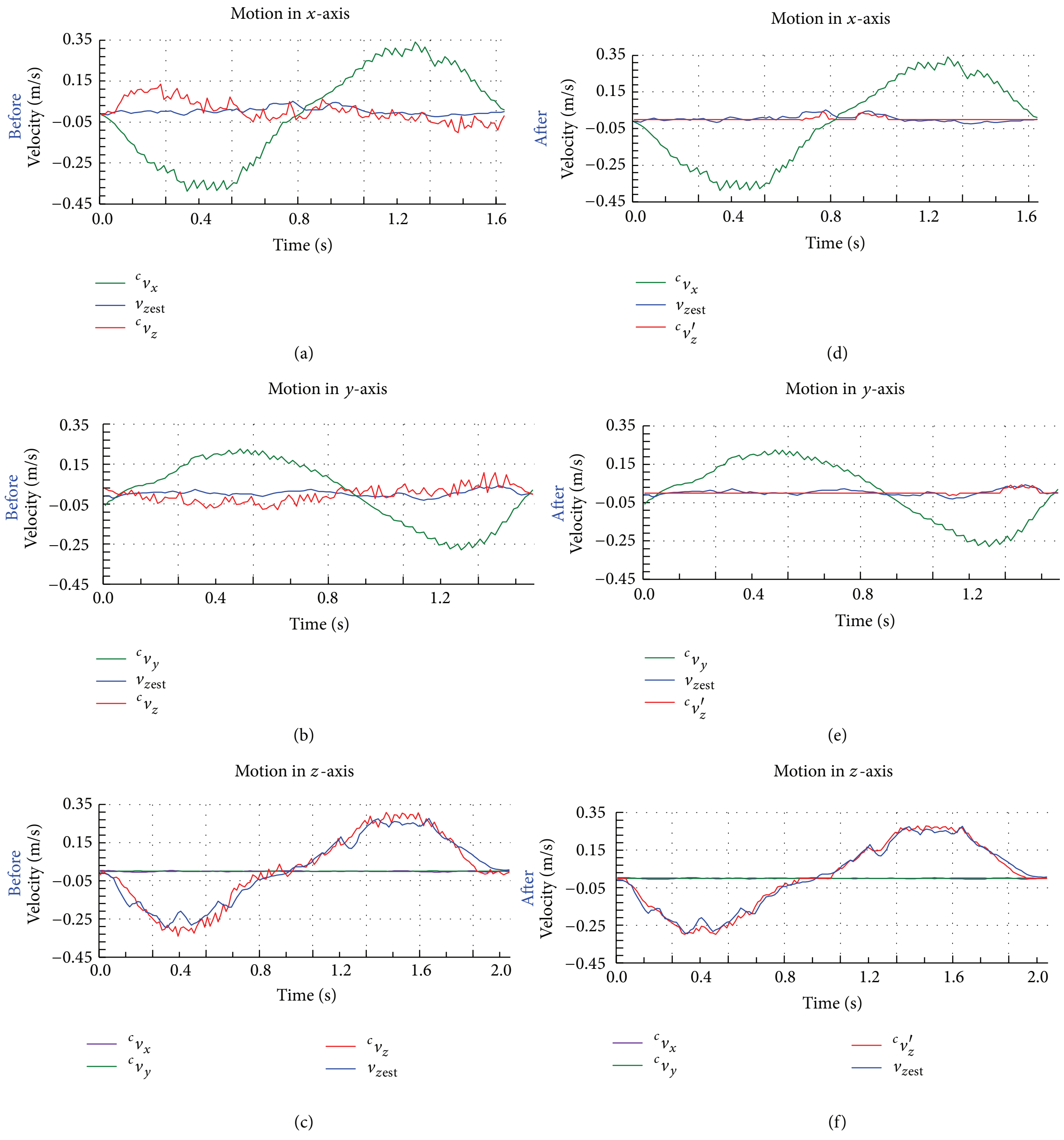
$$ G = \begin{bmatrix} 2\left(q_2'q_4' - q_1'q_3'\right) \\ 2\left(q_1'q_2' + q_3'q_4'\right) \\ \left(q_1'\right)^2 - \left(q_2'\right)^2 - \left(q_3'\right)^2 + \left(q_4'\right)^2 \end{bmatrix}, \tag{10} $$

$$ \left[a_{tx} \;\; a_{ty} \;\; a_{tz}\right] = {}^{b}A - G^{T}, \qquad {}^{b}A = \left[a_x \;\; a_y \;\; a_z\right], \tag{11} $$

$$ \left[0 \;\; a_{tx}' \;\; a_{ty}' \;\; a_{tz}'\right] = {}^{e}_{b}q^{*} \otimes \left[0 \;\; a_{tx} \;\; a_{ty} \;\; a_{tz}\right] \otimes {}^{e}_{b}q, \tag{12} $$

$$ {}^{e}A = \left[{}^{e}A\right] 9.81, \qquad {}^{e}A = \left[a_{tx}' \;\; a_{ty}' \;\; a_{tz}'\right]. \tag{13} $$

5.2. Combined Tracking. A block diagram of combined tracking is shown in Figure 10 which serves as second stage sensor fusion algorithm. A 9-state Kalman filter incorporated is the heart of this algorithm. The KF is an efficient recursive filter algorithm that provides optimal estimates of system states from noisy observation data given the underlying model of the system and assumes all the errors and measurements have zero mean white Gaussian noise. It is also well known that to



(a)

(d)

(b)

(e)

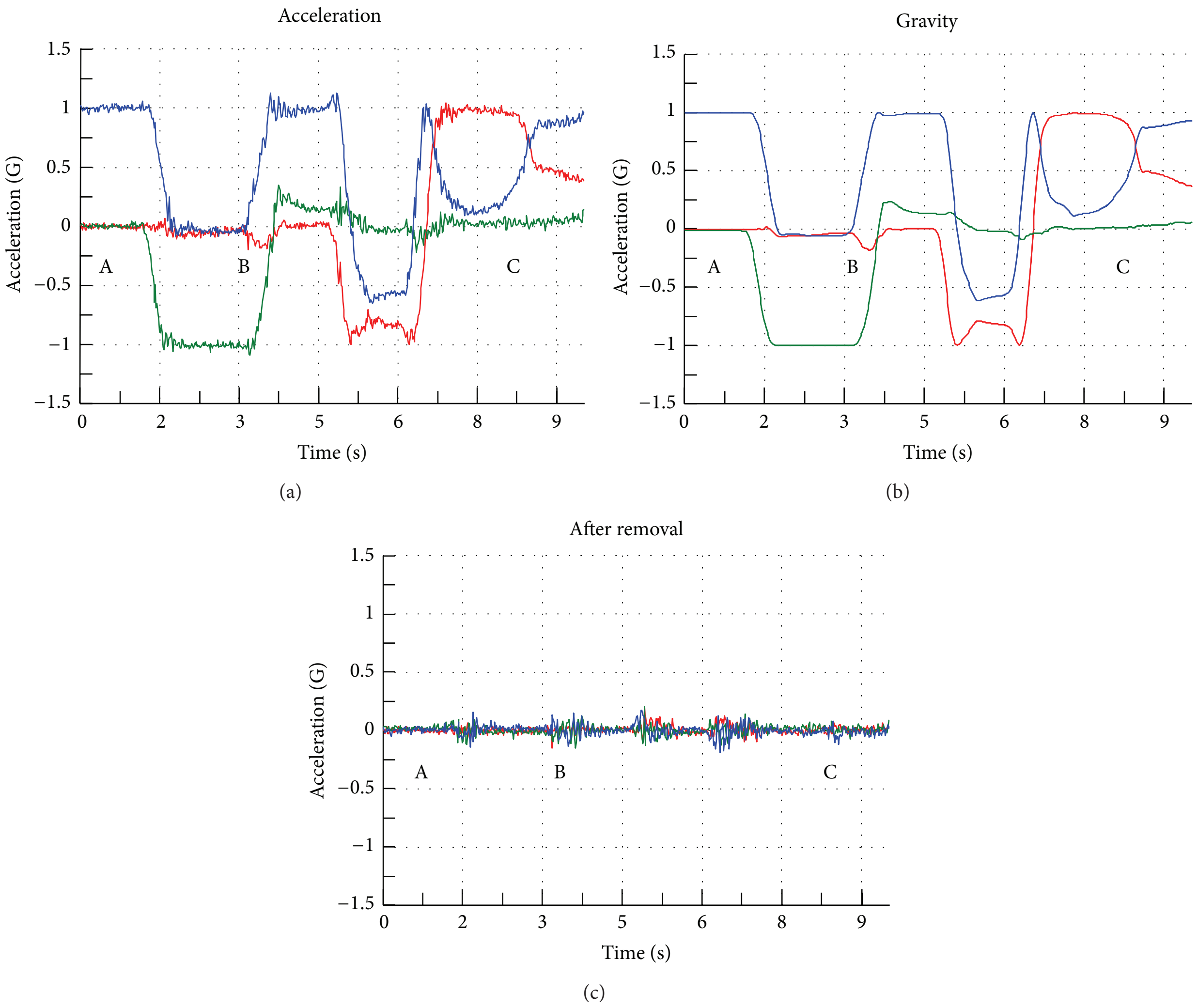
(c)

(f)

FIGURE 8: Before and after correction of ${}^{c}v_z$: (a) error introduced in ${}^{c}v_z$ when marker moved along x-axis; (d) after correcting error in ${}^{c}v_z$; (c) error introduced in ${}^{c}v_z$ when marker moved along y-axis; (e) after correcting error in ${}^{c}v_z$.

compensate for errors of inertial navigation systems, inertial sensors have to be assisted by other sensors, and use of a KF is common for fusing data from different sensors. Also KF is used in different applications, for example, moving target tracking in a video [31]. We propose fusion of high sampling rate MARG sensors and low frame rate vision sensor in second stage sensor fusion for 3D position tracking, which benefits from their complementary characteristics. The state variables of the KF are position, velocity, and translational acceleration.

The inertial navigation computing module estimates position and velocity from translational acceleration vector ${}^{e}A$ and corrections obtained from the filter at a previous time step. The KF takes these estimates and velocity vector



(a)

(b)

(c)

Figure 9: Gravity removal in acceleration: (a) acceleration; (b) gravity vector computed; (c) resulting translational acceleration after gravity removed.

${}^{c}V$ to find the optimal estimates for position, velocity, and acceleration. The filter architecture has the following system dynamics and measurement model:

$$\begin{aligned} X_k &= F_{k-1}X_{k-1} + w_{k-1}, \quad w_k \sim N(0, Q_k), \\ Z_k &= H_k X_k + v_k, \quad v_k \sim N(0, R_k). \end{aligned} \tag{14}$$

Time Update. The state estimate and error covariance are propagated based on the optimal estimation at previous time step $k-1$.

Consider the following:

$$\hat{x}_k = F x_{k-1}, \tag{15}$$

$$\hat{P}_k = F P_{k-1} F^T + Q. \tag{16}$$

For (15) we use estimates of inertial navigation computing task and optimal states at the previous time step $k-1$. The symbol (^) represents predictions, x_k is state vector, and F and Q are state transition and process noise covariance:

$$\begin{gathered} x_k = \begin{bmatrix} p_x & v_x & a_x & p_y & v_y & a_y & p_z & v_z & a_z \end{bmatrix}^T, \\ F = \begin{bmatrix} F_i & 0_3 & 0_3 \\ 0_3 & F_i & 0_3 \\ 0_3 & 0_3 & F_i \end{bmatrix}, \quad F_i = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad i = x, y, z, \\ Q = \begin{bmatrix} Q_i & 0_3 & 0_3 \\ 0_3 & Q_i & 0_3 \\ 0_3 & 0_3 & Q_i \end{bmatrix}, \quad Q_i = \begin{bmatrix} \frac{\Delta t^5}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} q_c. \end{gathered} \tag{17}$$

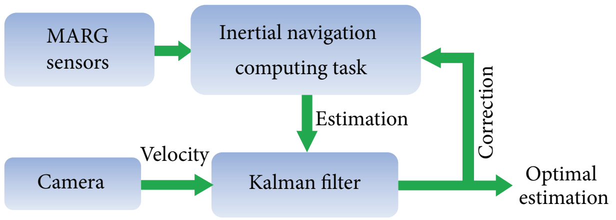


Figure 10: Combined tracking.

Matrices of size 9 × 9, 0_3 represent a 3 × 3 zero matrix, Δt is the sampling period of the MARG sensors, and q_c is the process noise covariance in continuous time.

Measurement Update. If measurements are available, this step incorporates those measurements in vector z_k, by adjusting the state variables, generating an optimal estimate and uncertainty P_k, using (18)–(20). K_k is the Kalman gain (9×9 matrix), often called acceptability vector of z_k

$$K_k = \widehat{P}_k H_k \left(H_k \widehat{P}_k H_k^T + R_k\right)^{-1}, \tag{18}$$

$$x_k = \widehat{x}_k + K_k \left(z_k - \widehat{x}_k\right), \tag{19}$$

$$P_k = \left(I - K_k H_k\right) \widehat{P}_k. \tag{20}$$

Measures of z_k, H_k, and R are 9 × 9 measurement sensitivity and noise covariance matrices, respectively. Covariance calculated using (20) will be used for the next iteration. Matrices z_k, H_k, and R_k are given below. Consider the following:

$$\begin{gathered} z_k = \begin{bmatrix} 0 & 0 & 0 & {}^cV & 0 & 0 & 0 \end{bmatrix}^T, \\ R_k = \begin{bmatrix} R_x & 0_3 & 0_3 \\ 0_3 & R_y & 0_3 \\ 0_3 & 0_3 & R_z \end{bmatrix}, \qquad R_i = I_{3\times 3}\sigma_i^2, \quad \sigma_i^2 = \text{variance}, \\ H_k = \begin{bmatrix} H_i & 0_3 & 0_3 \\ 0_3 & H_i & 0_3 \\ 0_3 & 0_3 & H_i \end{bmatrix}, \qquad H_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{gathered} \tag{21}$$

Since MARG sensors are sampled at 120 Hz and the camera is sampled at 75 FPS (<MARG sensors data rate), during each time step, we check the inertial navigation computing module for estimates and marker velocity vector ${}^cV$ from the camera if they are available. If both are available, both time and measurement update steps are updated to get the current optimal estimate. If both are not available, a previous optimal estimate becomes our current optimal estimate. If estimates are available and ${}^cV$ is not ready, only predictions are updated and these become the current optimal estimates. If estimates are not available but ${}^cV$ is ready, predictions and measurement updates are both calculated, but the prediction equations use previous optimal estimates since new estimates are not available from the inertial navigation module.

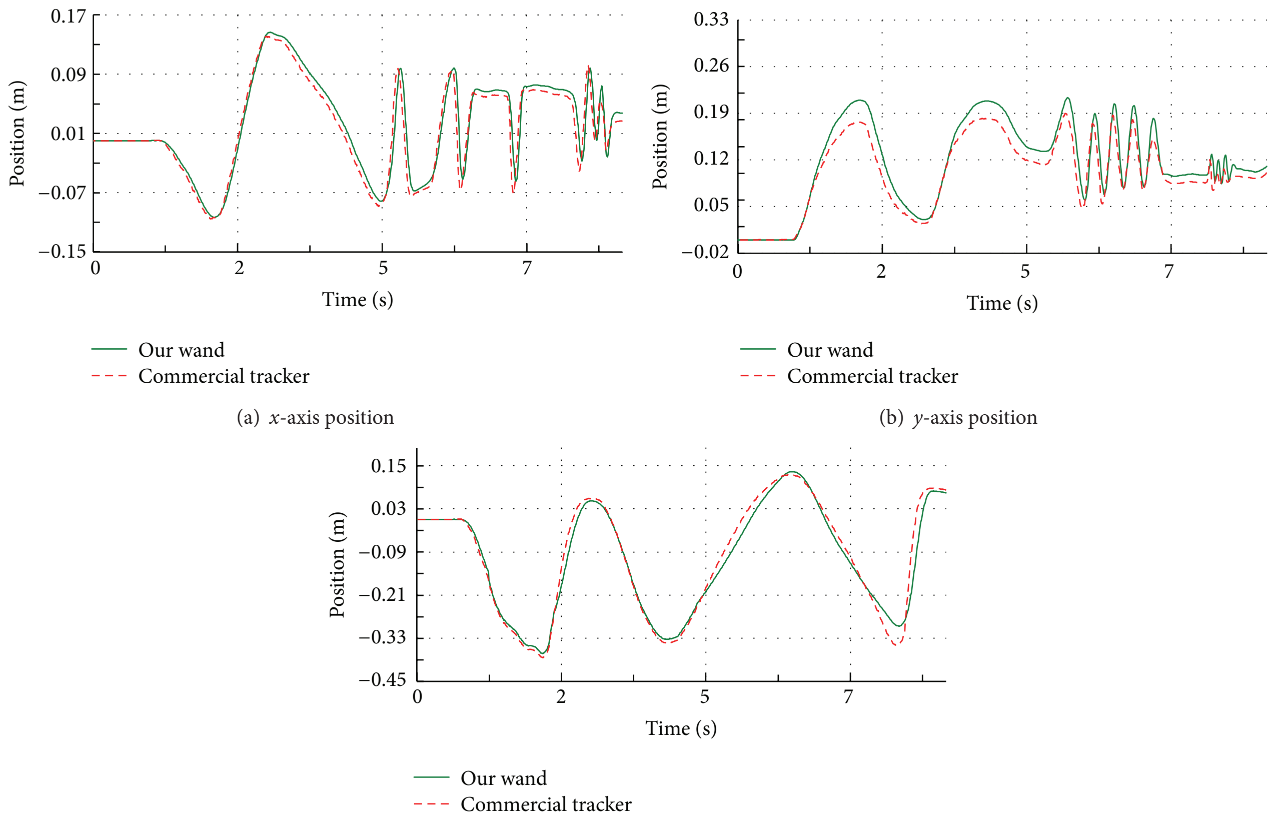
6. Results

The proposed tracking system was tested against a leading commercial DC magnetic tracking system at 120 Hz. To do so, we set the update rate of the proposed tracking system at 120 Hz and the camera FPS at 75 and fixed them to a rigid platform to be moved by hand. The trajectories of translational motion obtained from both the systems along x-, y-, and z-axes are plotted in Figures 11(a), 11(b), and 11(c), respectively. Comparison shows the potential of our system in both static and dynamic arbitrary movements. A close inspection of Figure 11(c) also reveals the linearity and accuracy of measurements obtained from (5), (6), and (7). This shows that our simple and efficient idea presented in Section 4 can find the marker velocity along the camera z-axis using area pixels of IR marker in thresholded image, without the need for another camera or additional markers to track the 3D position of the object.

In order to compare the quaternion orientation data of the two systems, orientation with respect to their fixed, steady state quaternions were measured and then decoupled to Euler parameters describing the pitch φ, roll θ, and heading ψ corresponding to rotations around the body frame x-, y-, and z-axes, respectively. Figure 12 shows plots of 3D orientation obtained from the wands complementary filter for which magnetic distortion compensation incorporated. To show the performance of our wand, comparison of the 3D trajectories for helical motion in the earth frame of reference is presented in Figure 13.

7. Conclusion

Motion tracking using MARG sensors with additional sensors is a mature field of research. Modern techniques [14–16] have focused on simpler fusion approaches on low power hardware to reach a wide range of users. We presented a simple and accurate approach for a wand system with two stage sensor fusion: the first stage of fusion offloads the 3D orientation computation from the computer, allowing focus on only 3D position computation in the computer as the camera is connected to it. The basic idea is to utilize low cost and wide field of view USB camera with IR filter to prevent numerical drift in the position computed from the acceleration of a MEMS accelerometer. Thus the overall system benefits from the complementary properties of inertial and vision sensing. Key advantages of the proposed system are (1) the working area of device, which allows user to interact with a computer or 3D TV at a comfortable distance by changing size of IR marker; (2) the small size and higher update rate; (3) the magnetic distortion compensation that helps to use the wand in challenging environments; and (4) another potential application of this device that includes air digital writing and signature verification.



(a) x-axis position

(b) y-axis position

(c) z-axis position

Figure 11: Comparison of trajectories of translational motion obtained from commercial tracker with our wand.

Appendix

Details of Complementary Filter

Complementary filter has two tuning parameters, proportional gain K_p and integral gain K_i. The K_p determines how quickly the algorithm output converges to the accelerometer and magnetometer measurements. In other words, K_p allows us to tune how much you trust sensors; a low value will trust the gyroscope more; for example, $K_p = 0$ means accelerometer and magnetometer data are ignored; $K_p = 0.5$ is suitable in most cases. The K_i corrects for gyroscope bias; $K_i = 0$ is used because calibrated data is supplied to the filter.

Quaternion Computation. Let

$$
{}^{e}_{b}q_t = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} q_{1,t-1} + \dfrac{1}{2}\left(-q_2\omega_x - q_3\omega_y - q_4\omega_z\right)\Delta_t \\ q_{2,t-1} + \dfrac{1}{2}\left(q_1\omega_x + q_{3,t-1}\omega_z - q_{4,t-1}\omega_y\right)\Delta_t \\ q_{3,t-1} + \dfrac{1}{2}\left(q_1\omega_y - q_{2,t-1}\omega_z + q_{4,t-1}\omega_x\right)\Delta_t \\ q_{4,t-1} + \dfrac{1}{2}\left(q_1\omega_z + q_{2,t-1}\omega_y - q_{3,t-1}\omega_x\right)\Delta_t \end{bmatrix}. \tag{A.1}
$$

Estimation Block. Equations (1)–(3) along with the following two equations will be computed in this block.

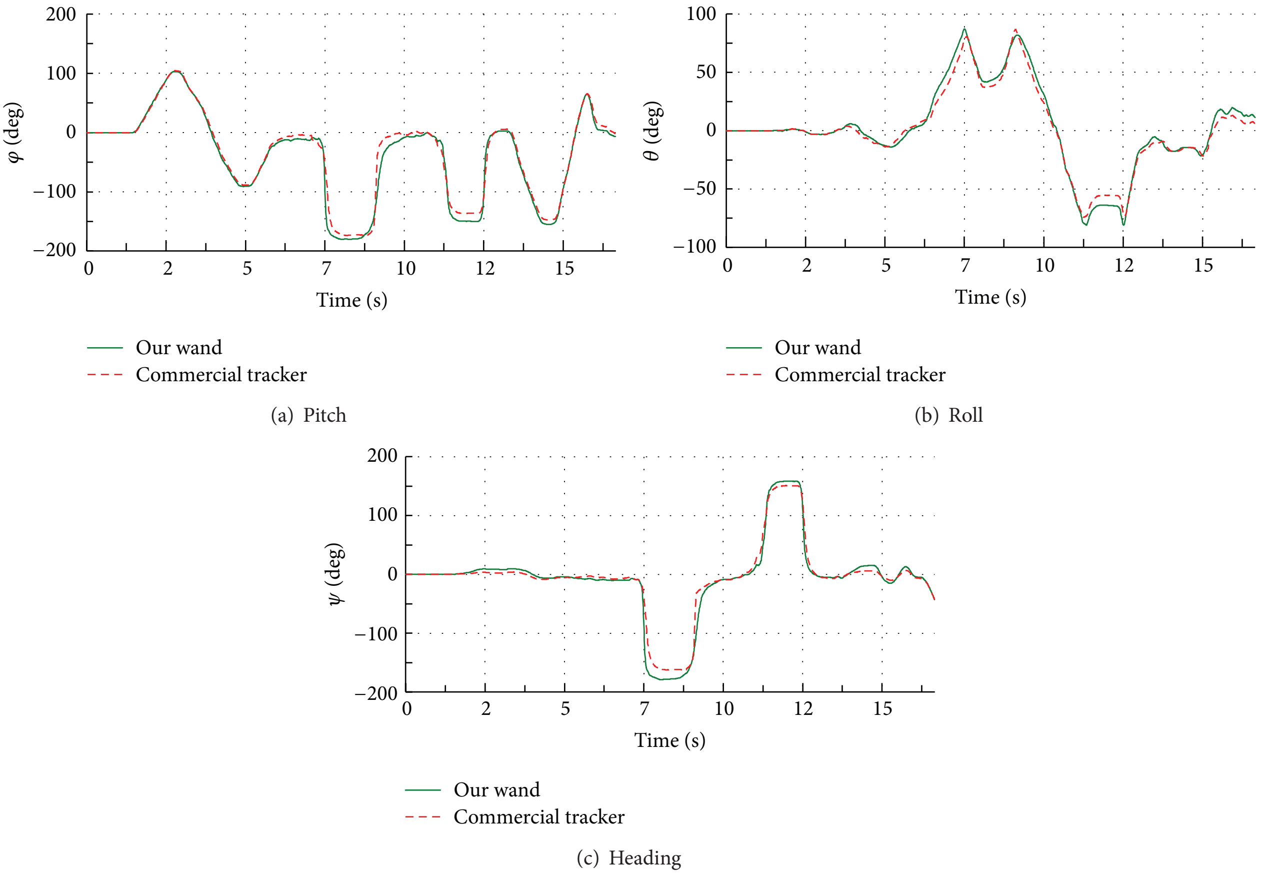
Estimated direction of gravity:

$$
\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \begin{bmatrix} 2\left(q_2q_4 - q_1q_3\right) \\ 2\left(q_1q_2 + q_3q_4\right) \\ q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix}. \tag{A.2}
$$

Estimated direction of magnetic field:

$$
\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} 2b_x\left(0.5 - q_3^2 - q_4^2\right) + 2b_z\left(q_2q_4 - q_1q_3\right) \\ 2b_x\left(q_2q_3 - q_1q_4\right) + 2b_z\left(q_1q_2 + q_3q_4\right) \\ 2b_x\left(q_1q_3 + q_2q_4\right) + 2b_z\left(0.5 - q_2^2 - q_3^2\right) \end{bmatrix}. \tag{A.3}
$$

Error Block. Error is computed by taking the cross product between estimated and measured quantities of gravity and magnetic field. The total error is the sum of gravity and magnetic field error vectors.



(a) Pitch

(b) Roll

(c) Heading

FIGURE 12: Comparison of Euler angles computed from quaternion orientations obtained from commercial tracker with our wand.

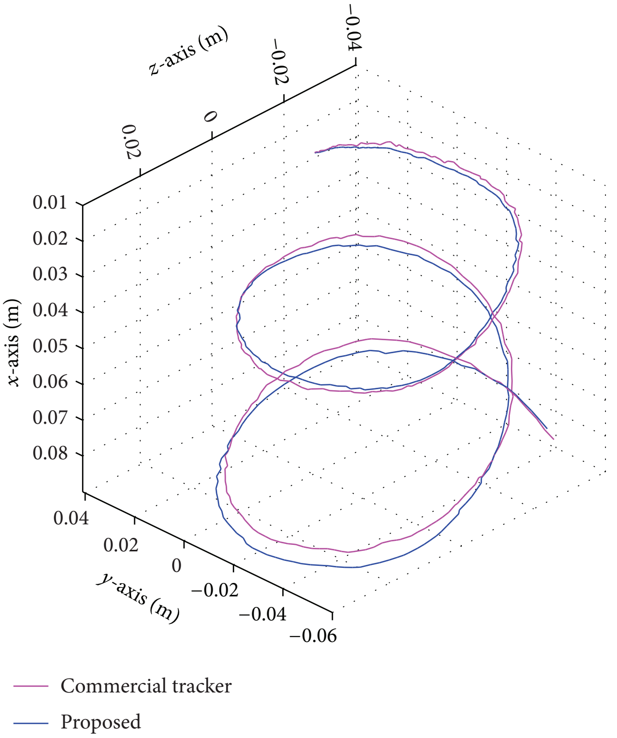


FIGURE 13: Comparison of 3D trajectories obtained from wand with commercial tracker.

Consider the following:

$$E = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = \begin{bmatrix} \left(a_{ny}v_z - a_{nz}v_y\right) + \left(m_{ny}w_z - m_{nz}w_y\right) \\ \left(a_{nz}v_x - a_{nx}v_z\right) + \left(m_{nz}w_x - m_{nx}w_z\right) \\ \left(a_{nx}v_y - a_{ny}v_x\right) + \left(m_{nx}w_y - m_{ny}w_x\right) \end{bmatrix}. \tag{A.4}$$

PI Controller. The following adjustment will be sent to the correction block where angular velocity ω of gyroscope will be adjusted:

$$U = \begin{bmatrix} k_P e_x + k_i i_x \\ k_P e_y + k_i i_y \\ k_P e_z + k_i i_z \end{bmatrix}, \qquad \text{where } \begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix} = I_t = \begin{cases} I_{t-1} + E * \Delta t & k_i > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{A.5}$$

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the Chung-Ang University Research Scholarship Grants in 2011.

References

[1] D. F. Keefe, R. C. Zeleznik, and D. H. Laidlaw, “Drawing on air: input techniques for controlled 3D line illustration,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 1067–1081, 2007.

[2] S. Schkolne, M. Pruett, and P. Schröder, “Surface drawing: creating organic 3D shapes with the hand and tangible tools,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’01)*, pp. 261–268, April 2001.

[3] S. Nam and Y. Chai, “SPACESKETCH: shape modeling with 3D meshes and control curves in stereoscopic environments,” *Computers and Graphics*, vol. 36, no. 5, pp. 526–533, 2012.

[4] H. Kim, D. Kim, J. Lee, and Y. Chai, “Real-time spatial surface modeling system usingwand traversal patterns of grid edges,” *IEICE Transactions on Information and Systems*, vol. E94-D, no. 8, pp. 1620–1627, 2011.

[5] L. B. Kara and K. Shimada, “Sketch-based 3D-shape creation for industrial styling design,” *IEEE Computer Graphics and Applications*, vol. 27, no. 1, pp. 60–71, 2007.

[6] D. H. Kim and M.-J. Kim, “A new modeling interface for the pen-input displays,” *CAD Computer Aided Design*, vol. 38, no. 3, pp. 210–223, 2006.

[7] G. Welch and E. Foxlin, “Motion tracking: no silver bullet, but a respectable arsenal,” *IEEE Computer Graphics and Applications*, vol. 22, no. 6, pp. 24–38, 2002.

[8] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV ’99)*, vol. 1, pp. 666–673, September 1999.

[9] R. P. McMahan, D. Gorton, J. Gresock, W. McConnell, and D. A. Bowman, “Separating the effects of level of immersion and 3D interaction techniques,” in *Proceedings of the 13th ACM Symposium Virtual Reality Software and Technology (VRST ’06)*, pp. 108–111, November 2006.

[10] A. Komlódi, E. Józsa, K. Hercegfi, S. Kucsora, and D. Borics, “Empirical usability evaluation of the Wii controller as an input device for the VirCA immersive virtual space,” in *Proceedings of the 2nd International Conference on Cognitive Infocommunications (CogInfoCom ’11)*, pp. 1–6, July 2011.

[11] J. Chun and S. Lee, “A vision-based 3D hand interaction for marker-based AR,” *International Journal of Multimedia and Ubiquitous Engineering*, vol. 7, no. 3, pp. 51–58, 2012.

[12] C. Rougier and J. Meunier, “3D head trajectory using a single camera,” *International Journal of Future Generation Communication & Networking*, vol. 3, no. 4, pp. 43–54, 2010.

[13] C. A. Wingrave, B. Williamson, P. D. Varcholik et al., “The wiimote and beyond: spatially convenient devices for 3D user interfaces,” *IEEE Computer Graphics and Applications*, vol. 30, no. 2, pp. 71–85, 2010.

[14] S. Kim, G. Park, S. Yim et al., “Gesture-recognizing hand-held interface with vibrotactile feedback for 3D interaction,” *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1169–1177, 2009.

[15] J.-S. Wang, Y.-L. Hsu, and J.-N. Liu, “An inertial-measurement-unit-based pen with a trajectory reconstruction algorithm and its applications,” *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3508–3521, 2010.

[16] R. Zhu and Z. Zhou, “A small low-cost hybrid orientation system and its error analysis,” *IEEE Sensors Journal*, vol. 9, no. 3, pp. 223–230, 2009.

[17] W.-C. Bang, W. Chang, K.-H. Kang, E.-S. Choi, A. Potanin, and D.-Y. Kim, “Self-contained spatial input device for wearable computers,” in *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC ’05)*, pp. 26–34, October 2005.

[18] A. D. Wu, E. N. Johnson, and A. A. Proctor, “Vision-aided inertial navigation for flight control,” *AIAA Journal of Aerospace Computing, Information and Communication*, vol. 2, no. 9, pp. 348–360, 2005.

[19] N. Parnian and F. Golnaraghi, “Integration of a multi-camera vision system and strapdown inertial navigation system (SDINS) with a modified Kalman filter,” *Sensors Journal*, vol. 10, no. 6, pp. 5378–5394, 2010.

[20] G. Pons-Moll, A. Baak, T. Helten, M. Müller, H.-P. Seidel, and B. Rosenhahn, “Multisensor-fusion for 3D full-body human motion capture,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’10)*, pp. 663–670, June 2010.

[21] V. Milanović, N. Siu, A. Kasturi, M. Radojičić, and Y. Su, “MEMSEye for optical 3D position and orientation measurement,” in *MOEMS and Miniaturized Systems X*, vol. 7930 of *Proceedings of SPIE*, 2011.

[22] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm,” in *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR ’11)*, pp. 1–7, July 2011.

[23] R. Mahony, T. Hamel, and J.-M. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[24] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, “A complementary filter for attitude estimation of a fixed-wing UAV,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’08)*, pp. 340–345, September 2008.

[25] A. M. Sabatini, “Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing,” *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.

[26] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, “An extended Kalman filter for quaternion-based orientation estimation using MARG sensors,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, pp. 2003–2011, November 2001.

[27] E. R. Bachmann, X. Yun, and C. W. Peterson, “An investigation of the effects of magnetic variations on inertial/magnetic orientation sensors,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA ’04)*, vol. 2, pp. 1115–1122, May 2004.

[28] A. Rahman, A. Salam, M. Islam, and P. Sarker, “An image based approach to compute object distance,” *International Journal of Computational Intelligence Systems*, vol. 1, no. 4, pp. 304–312, 2008.

[29] S. F. Hasan, R. M. N. Sadat, M. A. Rahman, and M. H. Kabir, “A precise and low complexity distance and size measurement of circular objects from camera position using still images,” in *Proceedings of the 4th International Conference on Electrical and Computer Engineering (ICECE ’06)*, pp. 439–442, December 2006.

[30] C. C. Tsang, G. C. T. Chow, P. H. W. Leong et al., “A novel real-time error compensation methodology for μIMU-based digital writing instrument,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO ’06)*, pp. 678–681, Kunming, China, December 2006.

[31] C. Suliman, C. Cruceru, and F. Moldoveanu, “Kalman filter based tracking in an video surveillance system,” *Advances in Electrical and Computer Engineering*, vol. 10, no. 2, pp. 30–34, 2010.