Real-Time Implementation of Orientation Correction Algorithm for 3D Hand Motion Tracking Interface

Nonnarit O-larnnithipong[1(✉)], Armando Barreto[1], Neeranut Ratchatanantakit[1], Sudarat Tangnimitchok[1], and Francisco R. Ortega[2]

[1] Electrical and Computer Engineering Department, Florida International University, Miami, FL 33199, USA
{nolar002, barretoa, nratc001, stang018}@fiu.edu
[2] School of Computer and Information Sciences, Florida International University, Miami, FL 33199, USA
fortega@cs.fiu.edu

Abstract. This paper outlines the real-time implementation of an orientation correction algorithm using the gravity vector and the magnetic North vector for a miniature, commercial-grade Inertial Measurement Unit to improve orientation tracking in 3D hand motion tracking interface. The algorithm uses the sensor fusion approach to determine the correct orientation of the human hand motion in 3D environment. The bias offset error is the IMU's systematic error that can cause a problem in orientation tracking called drift. The algorithm is able to determine the bias offset error and update the gyroscope reading to obtain unbiased angular velocity. Furthermore, the algorithm will compare the initial estimated orientation result by using other referencing sources which are the gravity vector measured from the accelerometer and the magnetic North vector measured from the magnetometer, resulting in the improvement of the estimated orientation. The orientation correction algorithm is implemented in real-time within Unity along with position tracking, through a system of infrared cameras. To validate the performance of the real-time implementation, the orientation estimated from the algorithm and the position obtained from the infrared cameras are applied to a 3D hand model. An experiment requiring the acquisition of cubic targets within a 3D environment using the 3D hand motion tracking interface was performed 30 times. Experimental results show that the algorithm can be implemented in real-time and can eliminate the drift in orientation tracking.

Keywords: Inertial Measurement Unit · Gyroscope drift
Orientation correction algorithm · Bias offset error
Quaternion correction using gravity vector and magnetic north vector
3D hand motion tracking interface

M. Antona and C. Stephanidis (Eds.): UAHCI 2018, LNCS 10907, pp. 228–242, 2018.
https://doi.org/10.1007/978-3-319-92049-8_17

1 Introduction

The design of natural human-computer interaction interfaces is becoming increasingly important to computer manufacturers. Researchers are proposing several ideas in order to create the interaction mechanisms that are as close as possible to the natural interaction between humans. The mouse is a common device that has been used to interact with personal computers for a long time. But it is a device that is hardly close to a natural human-computer interaction. The mouse is normally used to interact with a 2D computer screen, it will eventually have a limitation when it comes to the interaction in 3D virtual environments. Several related works [1–3] tried to overcome this limitation and found alternative mechanisms to interact with computers by using computer vision to determine hand gestures or using eye tracking to interact. In the modern day, the development of the 3D User Interfaces, especially for Virtual and Augmented Reality is increasingly emphasized. In order to effectively interact with immersive VR or AR environments, the systems should provide realistic visual and acoustic output [4–7].

We have verified that our orientation correction algorithm using gravity vector and magnetic North vector compensation can effectively be used to eliminate the gyroscope drift in commercial-grade Inertial Measurement Units. The IMU we used in our research is composed of different types of sensors; for example, accelerometers, magnetometers and gyroscopes. A gyroscope measures the angular velocity, which can be used to determine orientation by performing integration through time. When the IMU has no motion, the gyroscopes are supposed to output zero values as the measurements. But, for low-cost IMUs, they can output a non-zero value even when they are not moving. This systematic error within the IMUs is called the bias offset error. In determining the orientation by means of mathematical integration, this bias offset error accumulates through time resulting in an orientation error called drift. The drift is a major problem in the navigation systems that use low-cost IMUs to measure the orientation of moving parts [8, 9].

Many studies [10–12] use Kalman filters to eliminate gyroscope drift error in inertial sensors. Other studies [13, 14], have proposed the idea of estimating the orientation using other sensor fusion methods. They combine more than one type of sensor to calculate the orientation. This is because the Kalman filter approach can be complicated and hard to implement [15, 16]. The objective of this work is to develop a system capable of determining the movement of the human hand in real-time by combining two different sources of information: orientation tracking using Inertial Measurement Units (IMUs) and position tracking using infrared cameras. To achieve that, this research also proposes an algorithm to correct gyroscope drift within the inertial measurement unit. This will be done by estimating the gyroscope bias offset error during intervals when the sensor is static and using the gravity vector measured by the accelerometers and magnetic North vector measured by the magnetometer. The goal is to monitor the movement of the human hand, in three-dimensional space including translation and rotation, in real-time.

2 Methodology and Materials

The 3D hand motion tracking interface will become aware of hand position and orientation by determining the position of an infrared marker and data from the inertial measurement unit attached on a glove, respectively. The system consists of two main parts, which are: position tracking using OptiTrack V120: Trio and orientation tracking using Yost Labs 3-Space sensors.

2.1 Position Tracking Using OptiTrack V120:Trio

OptiTrack V120: Trio is the object tracking technology that consists of three infrared sensor units. Each unit has a circular array of 26 LEDs around an infrared camera. The OptiTrack will be used to track the position of the hand moving in 3D space. OptiTrack V120: Trio was attached to a stand, above the computer screen as shown in Fig. 1. Since three IR cameras in the system have fixed distances between each other, the OptiTrack V120: Trio was pre-calibrated from the manufacturer. The hardware can be conveniently connected to the computer using a USB cable. The device comes with the licensed software called Motive:Tracker, which is an engineering-grade rigid body and marker tracking system. IR-reflective dot markers are attached around the wrist of the glove as shown in Fig. 2 to act as the referencing points for other movements of the hand beyond the wrist. The intensity of the adjustable IR LEDs is reduced within Motive:Tracker in order to eliminate unwanted spurious reflections. Each of the dot markers attached around the wrist will be visible to the three IR cameras as a single point in 3D space as shown in Fig. 3. With the combination of image processing input from three infrared cameras, Motive:Tracker can compute the position of each detected marker and will continuously provide its Cartesian coordinates (position in x, y and z) in real-time. Motive:Tracker will also display a visualization of the marker in a 3D environment, as a single point. The software allows the user to transport the marker coordinates data to other application using NatNet SDK, capable of cross-platform data streaming. With NatNet SDK, a console application written in C++ was created to stream the marker coordinates detected in Motive:Tracker. This console application establishes the connection to a NatNet server in Motive:Tracker, it receives a data stream and encodes it to an XML document. The application outputs XML locally over UDP to Unity 5 as shown in Fig. 4. In Unity, scripts written in C# were created to receive the XML document via the UDP connection. The script will be able to parse the tracking data from the streaming and apply the translation to the GameObject (3D Hand model shown in Fig. 5), which was previously prepared for the visualization of the hand motion tracking system.

2.2 Orientation Tracking Using Yost Labs 3-Space Sensors

In this work, three Yost Labs 3-Space sensors are used to determine the orientations of the hand, proximal phalange and middle phalange of the index finger. The orientation of the distal phalange of the index finger is not measured with a sensor, instead it is calculated based on the angle of middle phalange as described below. This type of low-cost Inertial Measurement Unit contains three different types of inertial sensors,

Fig. 1. The setup of OptiTrack V120: Trio with Motive:Tracker software

Fig. 2. The glove with IR-reflective dot markers attached on the wrist, one Yost Labs 3-space sensor attached on the back of the hand, and two Yost Labs 3-space sensors attached on the index finger

which are accelerometer, gyroscope and magnetometer. Each of the sensing units is able to measure in three orthogonal axes. The sensor can provide several types of information for the inertial measurement such as linear acceleration, direction of north magnetic field and gyroscope data (angular velocity). Three IMUs are connected to the host PC via USB cables and a specific command has to be sent to the sensor in order to receive the desired information. In Unity, a script written in C# was created to receive the streaming data from the Yost Labs 3-Space sensors. The script will parse the data into three sets of 3-dimensional vector: acceleration, angular velocity and direction of the magnetic field. The C# script then applies the orientation correction algorithm using gravity vector and magnetic North vector compensation to calculate an estimated output quaternion and apply the rotation to the same 3D hand model described in the

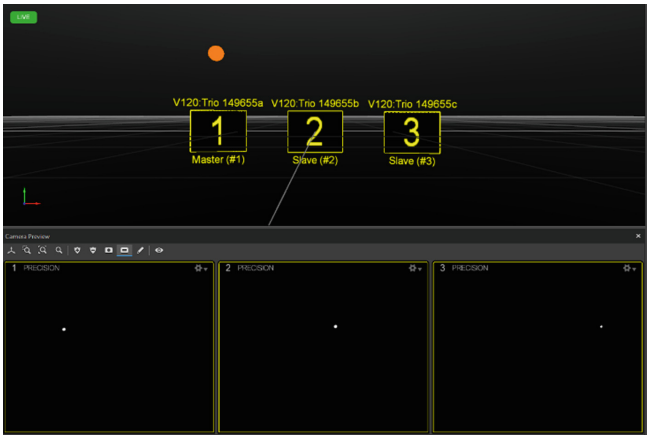

Fig. 3. Visible dot marker on three infrared cameras (white dots) and position of the marker in 3D space shown as an orange dot. (Color figure online)

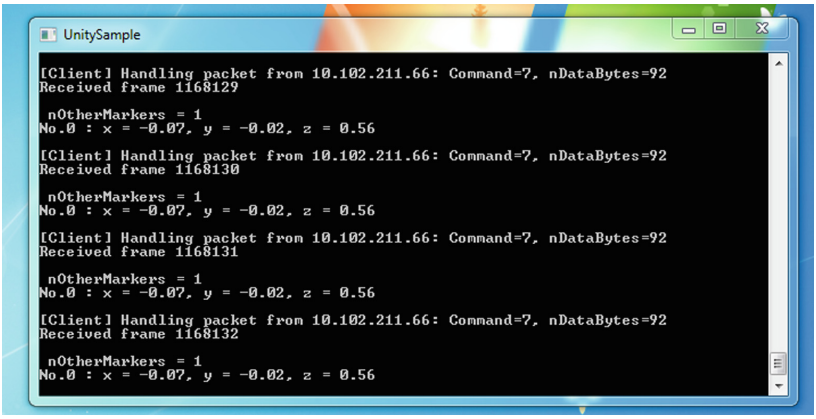

Fig. 4. Console application to transport position marker data from Motive:Tracker to Unity

discussion of the position tracking (above). As mentioned earlier about the orientation of the distal phalange of the index finger, an empirical study and experimental observations [17] found the approximate dependency of the joint angle of the distal phalange and the joint angle of the middle phalange to be as shown in Eq. (1)

$$\theta_{Distal} = \frac{2}{3}\theta_{Middle} \tag{1}$$

Bias Offset Estimation. Bias offset error is one of the systematic errors within inertial measurement units. When the IMU is not moving or rotating, the IMU should provide the reading value of zero. But for low-cost IMUs, the bias offset error occurs and causes

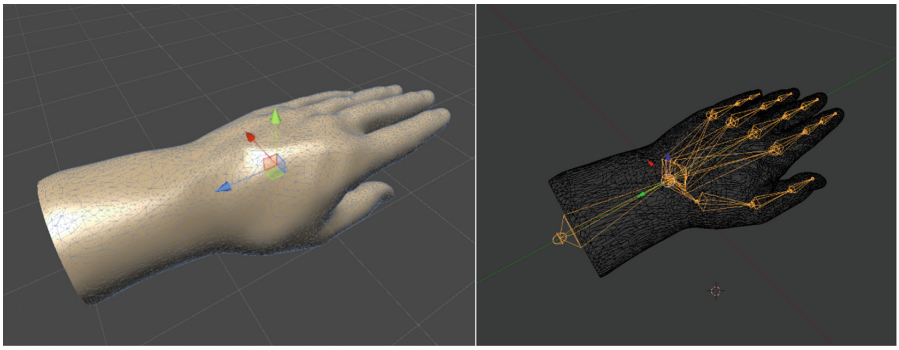

Fig. 5. 3D Hand model in Unity 5

a major problem in inertial measurement called drift. To determine the bias offset error in a real-time manner, the new bias offset error will be calculated only when the value of angular velocity is less than a predefined threshold for 5 consecutive samples (representing an interval of 250 ms). The bias offset error is calculated by determining the arithmetic mean of 5 consecutive angular velocity samples. The calculated bias offset error will be subtracted from the raw gyroscope data resulting in unbiased angular velocity as shown in Eq. (2).

$$\omega_{\mathrm{B}} = \vec{\omega}_0 - \hat{\mathrm{b}} \tag{2}$$

In this project, the quaternion notation is used to represent rotations because it is a common notation used for 3D rotation in Unity and avoids a measurement problem known as "gimbal lock". The quaternion rate ($\dot{q}$) is calculated using unbiased angular velocity as shown in Eq. (3).

$$\dot{q} = \frac{1}{2}\hat{q}_0 \otimes \omega_B \tag{3}$$

$$q_G = \exp\left((\Delta t)\dot{q} \otimes \hat{q}_0^*\right) \otimes \hat{q}_0 \tag{4}$$

In Eq. (4), the quaternion (q_G) is calculated by using quaternion rate ($\dot{q}$) obtained from Eq. (3), the previous quaternion ($\hat{q}_0$) and the sampling time (Δt). This quaternion (q_G) can be used to describe the orientation of the object in 3D space. The quaternion (q_G) will be further processed by the quaternion correction process using the gravity vector and the magnetic North vector.

Quaternion Correction

Using the Gravity Vector. When the IMU is static, each of the three axes of the accelerometer will measure only the acceleration due to gravity. The gravity vector always points towards the Earth's center in the Earth frame. If the sensor is in an oblique orientation, the acceleration due to gravity will be decomposed into three orthogonal axes resulting in gravity vector measured in the sensor's frame, which

describes the inclination of the sensor compared to the vertical gravity vector in the Earth frame. We can use quaternion (q_G) calculated in Eq. (3) to rotate [18] the initial gravity vector in the Earth frame (A_{int}) into the gravity vector in the sensor frame called *calculated gravity vector* ($\vec{a}(q_G)$) as shown in Eq. (5).

$$\vec{a}(q_G) = q_G^* \otimes A_{int} \otimes q_G \tag{5}$$

If there is no error occurred in the quaternion (q_G), the calculated gravity vector ($\vec{a}(q_G)$) will match the *measured gravity vector* ($\vec{a}_0$) from accelerometer readings. Otherwise, the angular difference between these two vectors in quaternion form (Δq_A) will be calculated and used to correct the quaternion (q_G) as shown in Eq. (6).

$$\Delta q_A = \mathcal{H}(\vec{q}_{Av}, q_{Aw}) \tag{6}$$

$$\vec{q}_{Av} = \vec{a}_0 \times \vec{a}(q_G) \tag{7}$$

$$q_{Aw} = \|\vec{a}_0\|\|\vec{a}(q_G)\| + \vec{a}_0 \cdot \vec{a}(q_G) \tag{8}$$

$$\hat{q}_{GA} = q_G \otimes \Delta q_A \tag{9}$$

The orientation quaternion corrected the using gravity vector, denoted by $\hat{q}_{GA}$, is calculated by post-multiplying Δq_A to the quaternion (q_G) calculated from Eq. (4)

Using the Magnetic North Vector. In a way similar to the decomposition of the gravity vector measured by accelerometer described above, the magnetic North vector measured by magnetometer will be decomposed into three orthogonal axes resulting in the magnetic North vector measured in the sensor's frame. This describes the current inclination of the sensor compared to the direction of the magnetic North vector in the Earth frame. We can use quaternion (q_G) calculated in Eq. (3) to rotate the initial magnetic North vector in the Earth frame (M_{int}) into the magnetic North vector in the sensor frame called *calculated magnetic North vector* ($\vec{m}(q_G)$) as shown in Eq. (10).

$$\vec{m}(q_G) = q_G^* \otimes M_{int} \otimes q_G \tag{10}$$

If there is no error in the quaternion (q_G), the calculated magnetic North vector ($\vec{m}(q_G)$) will match the *measured magnetic North vector* ($\vec{m}_0$) from magnetometer readings. Otherwise, the angular difference between these two vectors in quaternion form (Δq_M) will be calculated and used to correct the quaternion (q_G) as shown in Eq. (11).

$$\Delta q_M = \mathcal{H}(\vec{q}_{Mv}, q_{Mw}) \tag{11}$$

$$\vec{q}_{Mv} = \vec{m}_0 \times \vec{m}(q_G) \tag{12}$$

$$q_{Mw} = \|\vec{m}_0\|\|\vec{m}(q_G)\| + \vec{m}_0 \cdot \vec{m}(q_G) \tag{13}$$

$$\hat{q}_{GM} = q_G \otimes \Delta q_M \tag{14}$$

The orientation quaternion corrected using the magnetic North vector, denoted by $\hat{q}_{GM}$, is calculated by post-multiplying Δq_M to the quaternion (q_G) calculated from Eq. (4). Note that regardless of the sensor's motion, the magnetometer will always measure the direction of the magnetic North vector (assuming that the magnetic field is constant in the testing area), unlike the accelerometer, which will include linear acceleration in the measurement when the sensor is in motion.

Quaternion Interpolation. Quaternion Interpolation is the parametric function that can interpolate the intermediate rotation between two quaternions by giving a control parameter that ranges from 0 to 1. In [19], even though the bias offset estimation and quaternion correction using gravity vector has been used, the resulting estimated orientation is not as expected. This is because when the sensor is in motion, the accelerometer measures not just only acceleration due to gravity but also measures the acceleration from linear motion. By using quaternion interpolation, we can control whether the output estimated orientation will be depending more on the accelerometer or magnetometer as shown in Eq. (15).

$$\hat{q}_{OUT} = \frac{\hat{q}_{GM} \sin((1-\alpha)\Omega) + \hat{q}_{GA} \sin(\alpha\Omega)}{\sin(\Omega)} \tag{15}$$

$$\cos(\Omega) = \hat{q}_{GM} \cdot \hat{q}_{GA} \tag{16}$$

In Eq. (15), it is one of the several types of quaternion interpolation called Spherical Linear Interpolation (SLERP) [20], which includes the control parameter (α). This parameter represents the "stillness" of the module, from 0 to 1 (1 = no movement). If α is approaching zero, it indicates that the sensor is in rapid motion and the output quaternion ($\hat{q}_{OUT}$) will tend to depend more on the quaternion correction using the magnetometer. If α is approaching one, it means that the sensor is in a static period and the output quaternion ($\hat{q}_{OUT}$) will tend to depend more on the quaternion correction using the gravity vector. The orientation correction algorithm using gravity vector and magnetic North vector is also visually described in Fig. 6.

3 Implementation

3.1 Creating the Unity Game Scene

In order to prove the concept of real-time implementation of the orientation correction algorithm using gravity vector and magnetic North vector for the hand motion tracking interface, the algorithm described in Sect. 2.2 (Orientation Tracking using Yost Lab 3-Space sensor) was adapted for real-time performance using a C# script within Unity. To evaluate the compatibility of this orientation correction algorithm on the hand motion tracking interface, a game scene in Unity has been created as shown in Fig. 7.

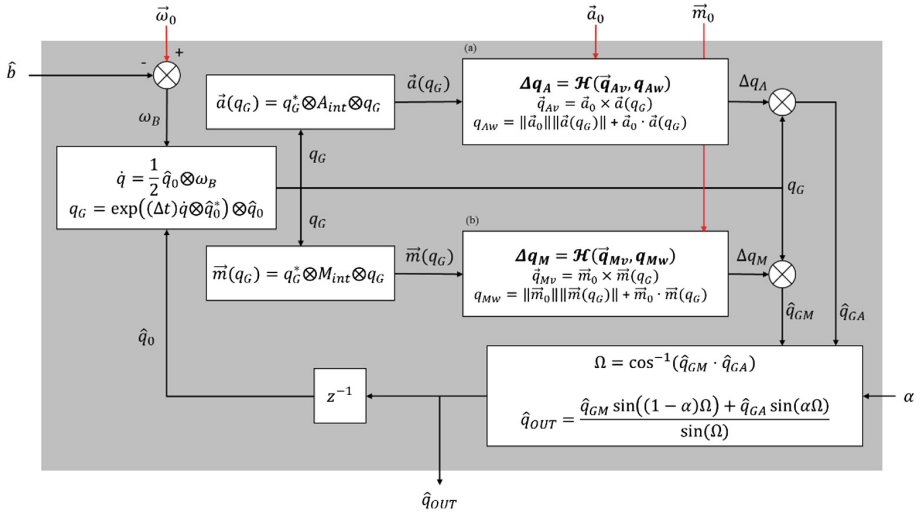

Fig. 6. Block diagram of the adopted orientation correction algorithm using gravity vector and magnetic North vector correction

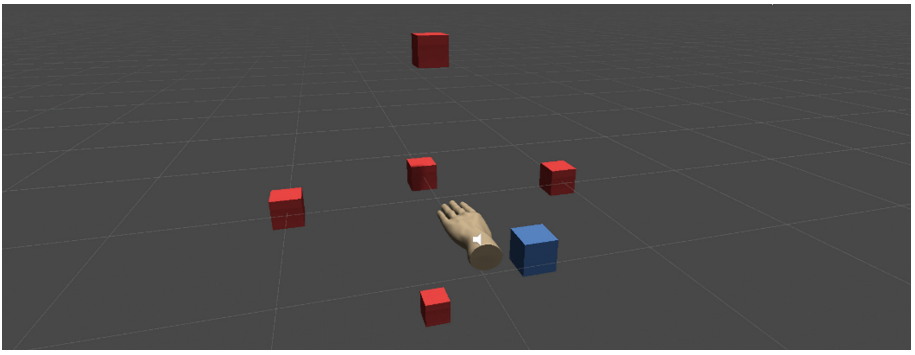

Fig. 7. Unity game scene for testing real-time implementation of the orientation correction algorithm

The 3D hand model shown in Fig. 8 was attached to the C# scripts that stream the marker position in 3D space and raw accelerometer, gyroscope and magnetometer data. The marker position from OptiTrack V120:Trio were assigned as the position of the 3D hand model. The C# script that streams raw accelerometer, gyroscope and magnetometer data also implemented the orientation correction algorithm using gravity vector and magnetic North vector. For every frame of rendering, the output estimated quaternion ($\hat{q}_{OUT}$) was calculated for the hand, proximal phalange, middle phalange and distal phalange. The quaternions were assigned as the rotations for the respective

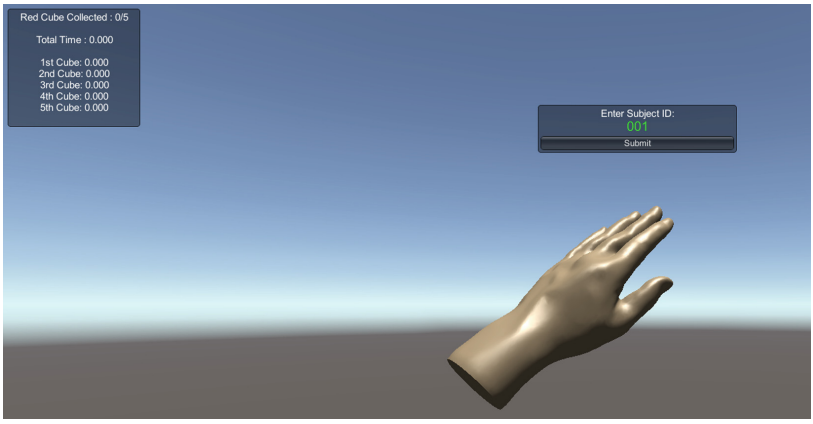

Fig. 8. Initial stage of the play mode when the subject ID is asked

parts of the hand. By combining the position tracking and orientation tracking to the 3D hand model, we can accomplish the goals of the hand motion tracking interface.

3.2 Evaluating the Hand Motion Tracking Interface Performance

To evaluate the performance of the hand motion tracking interface, a game scene with five red cubes and one blue cube was created. The experimental subject wore the glove and was asked to perform a task to "acquire" the red cubes in 3D space. A single red cube will appear in the scene after acquiring the blue cube. The acquisition of the blue cube marks the starting time for the subject to acquire the red cube. The red cubes are placed in 3D positions that are equal in distances from the origin. The time to acquire

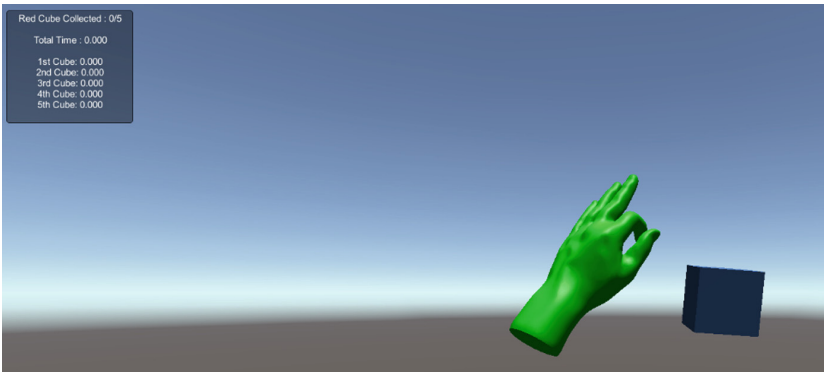

Fig. 9. The 3D hand model will turn into green indicating the state of flexing (Color figure online)

each red cube was recorded. The blue cube appeared for the first time after the subject entered the Subject ID as shown in Fig. 8.

In order to acquire each of the cubes in this task, the subject has to flex his/her index finger while colliding with the cube. The 3D hand model will change its color into green when the flexion of the index finger is detected, as shown in Fig. 9.

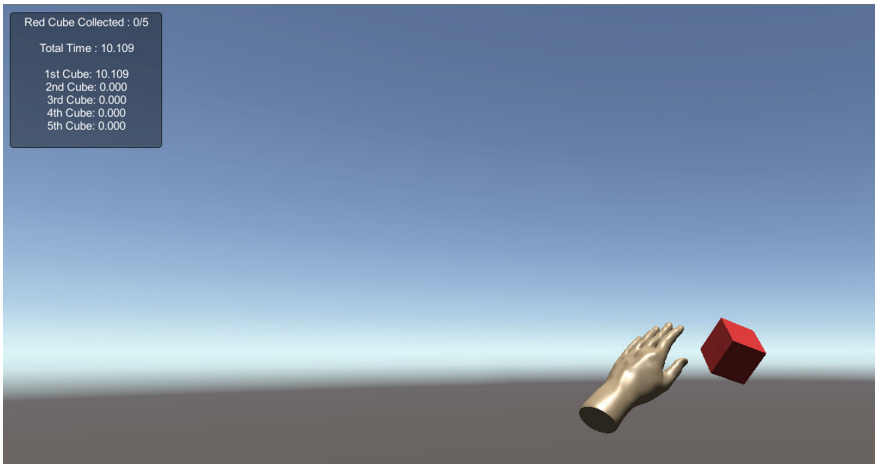

Fig. 10. The red cube will appear after acquiring the blue cube (Color figure online)

In every trial, after the blue cube is acquired, the red cube will appear. The subject will try to complete the trial by moving his/her hand in 3D space to reach the red cube and flex the index finger to acquire it as shown in Fig. 10. After the subject completes a trial by acquiring each of the 5 red cubes, the trial time will be recorded. When all 5 cubes have been acquired, the total experiment time will be calculated as the sum of the 5 trial times.

4 Results and Discussion

4.1 Static Test

To perform the static test, all three inertial measurement units were fixed to a table, to prevent them from moving. The output estimated quaternions for the orientation of hand, index proximal phalange and index middle phalange were recorded for 5 min. The output estimated quaternions were recorded when the orientation correction algorithm using gravity vector and magnetic North vector are both disabled and enabled. In Fig. 11, the output estimated quaternions for hand, index proximal phalange and middle phalange recorded while the orientation correction algorithm was disabled are shown. The result indicates that the bias offset error causes the angular measurement to drift even though the sensors were placed statically on the table. It was found that the drifts occurred in different rates among the three inertial measurement

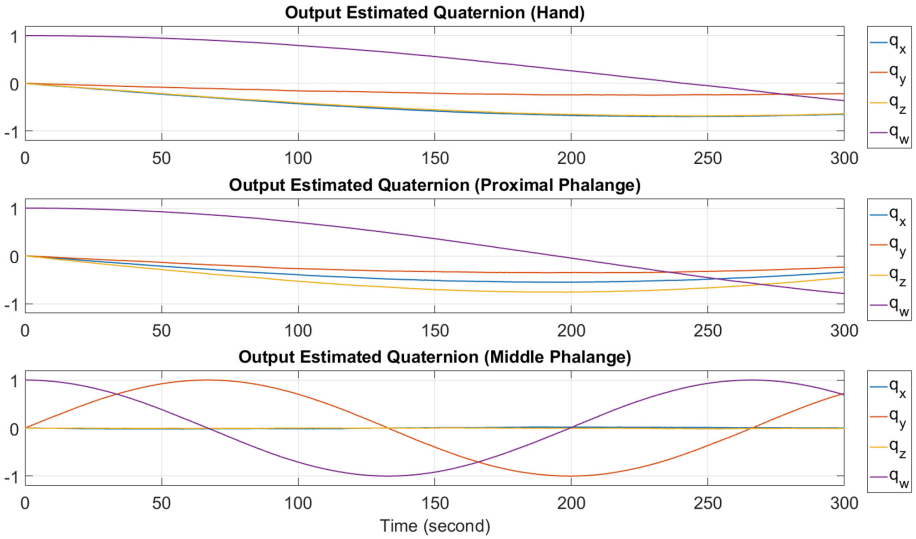

Fig. 11. Output estimated Quaternions without orientation correction algorithm

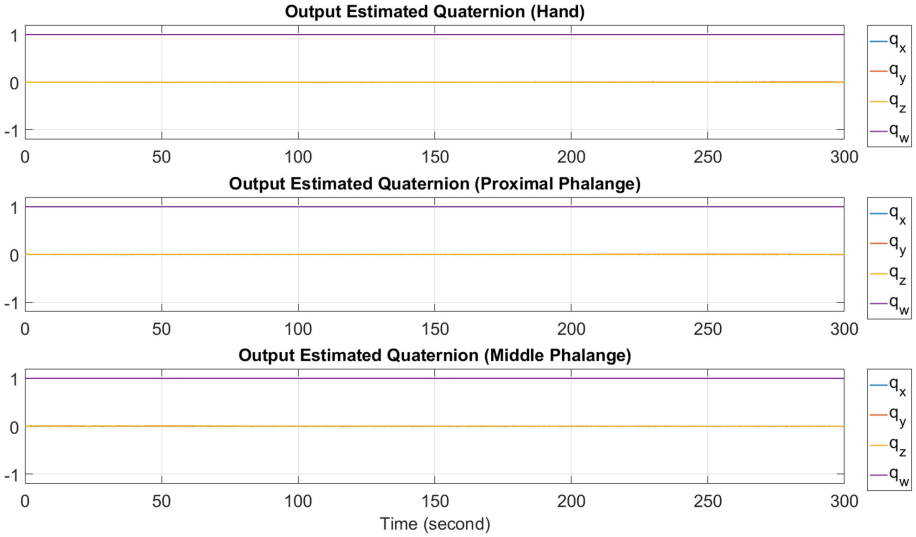

Fig. 12. Output estimated Quaternions with orientation correction algorithm

units. This is because each sensor has a different bias offset error. After enabling the orientation correction algorithm, the plots in Fig. 12 representing the orientation of 0° angle show that the orientation correction algorithm can improve the estimated quaternion in all three inertial measurement units. This confirmed that in the static mode (when there is no motion applied to the sensors) the orientation correction

algorithm using gravity vector and magnetic North vector effectively improve the orientation measurement using the inertial measurement units in real-time.

Table 1. Statistical data of the time used to acquire red cubes

Statistic values	Time in seconds					
	1st cube (front)	2nd cube (left)	3rd cube (right)	4th cube (up)	5th cube (down)	Total time
Mean	3.34	3.44	5.04	4.60	7.33	23.75
SD	1.04	1.07	1.83	2.01	4.12	6.38
Min	2.05	1.48	2.00	1.38	2.52	15.27
Max	6.45	5.59	8.90	8.77	20.87	36.02

4.2 Performing the Experimental Task Using the Hand Motion Tracking Interface

To evaluate the performance of the hand motion tracking interface, 30 experiments (each consisting of the acquisition of 5 red cubes) were performed in the 3D environment (described above). During these experiments, the orientation correction algorithm using gravity vector and magnetic North vector was enabled. The time used to acquire each cube was recorded. The statistical characteristics of these acquisition time are shown in Table 1.

From the results shown in Table 1, it can be seen that the 5 acquisition times in each experiment added to 23.75 s on average, with the standard deviation of 6.38 s. The minimum total time was 15.27 s, whereas the maximum total time was 36.02 s. The red cube that took the longest time to acquire (on average) was the 5th cube, which appeared at the bottom of the screen. It seems, then, that the 5th red cube was more difficult to reach, compared to the other red cubes because it's almost out of the OptiTrack V120:Trio field of view. From the experimental data, it can be verified that the real-time implementation of the orientation correction algorithm using gravity vector and magnetic North vector can be effectively applied with the hand motion tracking interface.

5 Conclusion

As verified by the results, we found that we are able to implement the orientation correction algorithm using gravity vector and magnetic North vector compensation in a real-time manner. Our approach is able to correct the drift in the gyroscope measurements. This method will be one of the effective approaches for the orientation tracking in 3D hand motion tracking interface which can be an alternative way to achieve interactions between a human and a computer. This can also be a significant contribution to improvement in the realism of natural human-computer interactions.

Acknowledgment. This research was supported by National Sciences Foundation grants HRD-0833093 and CNS-1532061 and the FIU Graduate School Dissertation Year Fellowship awarded to Mr. Nonnarit O-larnnithipong.

References

1. Zhang, X., Liu, X., Yuan, S., Lin, S.: Eye tracking based control system for natural human-computer interaction. Comput. Intell. Neurosci. **2017**, 9 p. (2017). Article ID 5739301. https://doi.org/10.1155/2017/5739301
2. Roh, M., Kang, D., Huh, S., et al.: A virtual mouse interface with a two-layered Bayesian network. Multimedia Tools Appl. **76**(2), 1615–1638 (2017)
3. Pavlovic, V.I., et al.: Visual interpretation of hand gestures for human-computer interaction: a review. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 677–695 (1997)
4. Mccall, R., et al.: Measuring presence in virtual environments. ACM (2004)
5. Slater, M., Usoh, M., Steed, A.: Depth of presence in virtual environments. Presence: Teleoper. Virtual Environ. **3**(2), 130–144 (1994)
6. Slater, M., Wilbur, S.: A framework for immersive virtual environments (FIVE). Presence: Teleoper. Virtual Environ. **6**(6), 603 (1997)
7. Heeter, C.: Being there: the subjective experience of presence. Presence: Teleoper. Virtual Environ. **1**(2), 262–271 (1992). https://doi.org/10.1162/pres.1992.1.2.262
8. Sukkarieh, S., Nebot, E.M.: A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. IEEE Trans. Robot. Autom. **15**(3), 572 (1999)
9. Borenstein, J., et al.: Mobile robot positioning sensors and techniques. In: Naval Command Control and Ocean Surveillance Center RDT and E Div, San Diego, CA (1997)
10. Marins, J.L., Yun, X., Bachmann, E.R., et al.: An extended Kalman filter for quaternion-based orientation estimation using MARG sensors. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol 4, pp. 2003–2011. IEEE (2001)
11. Yun, X., Bachmann, E.R.: Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking. IEEE Trans. Robot. **22** (6), 1216–1227 (2006). https://doi.org/10.1109/TRO.2006.886270
12. Yun, X., Lizarraga, M., Bachmann, E.R., et al.: An improved quaternion-based Kalman filter for real-time tracking of rigid body orientation. In: Intelligent Robots and Systems, 2003. IEEE/RSJ International Conference, vol 2, pp. 1074–1079. IEEE (2003)
13. Bachmann, E.R., Duman, I., Usta, U.Y., et al.: Orientation tracking for humans and robots using inertial sensors. In: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA 1999, pp. 187–194. IEEE (1999)
14. Kong, X.: INS algorithm using quaternion model for low cost IMU. Robot. Auton. Syst. **46** (4), 221–246 (2004)
15. Madgwick, S.O., Harrison, A.J., Vaidyanathan, R.: Estimation of IMU and MARG orientation using a gradient descent algorithm. In: IEEE International Conference on Rehabilitation Robotics (ICORR), pp. 1–7. IEEE (2011)
16. Madgwick, S.: An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io, University of Bristol (UK), p. 25 (2010)
17. Ip, H.H., Chan, C.S.: Dynamic simulation of human hand motion using an anatomically correct hierarchical approach. In: IEEE International Conference on Anonymous Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, vol. 2, pp. 1307–1312. IEEE (1997)

18. Kuipers, J.B.: Quaternions and Rotation Sequences: A Primer With Applications to Orbits, Aero-Space, and Virtual Reality. Princeton University Press, Princeton (1999)
19. O-larnnithipong, N., Barreto, A.: Gyroscope drift correction algorithm for inertial measurement unit used in hand motion tracking. In: IEEE SENSORS 2016, pp. 1–3 (2016)
20. Dam, E.B., Koch, M., Lillholm, M.: Quaternions, Interpolation and Animation. Datalogisk Institut, Kbenhavns Universitet, Copenhagen (1998)