International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-1, November 2019

Inertial Tracking Systems for Virtual Reality Devices

Sergei Sergeevich Smirnov, Dmitrii Aleksandrovich Sytnik, Nikolai Valerievich Bradis, Andrei Ivanovich Kardakov

Abstract*: Tracking systems are the most important components in virtual reality devices and appliances. In the simplest case, in order to represent current picture for a user of virtual reality glasses, it is required to provide head tracking. In complex devices of virtual reality, it is also required to track position of device itself and dynamics of its motions. This work is aimed at development of 3D positioning algorithms for virtual reality devices and appliances. In this work positioning is based on operation of inertial sensors. The issues of motion capture and calibration of inertial devices are considered as well as application of Madgwick filter for noise suppression and quaternions for description of coordinate rotation. This work discusses the application of the developed algorithms for various virtual reality devices and appliances.

Keywords* : Sphere, Virtual reality, Madgwick filter, Gyroscope, Accelerometer, Quaternion.

I. INTRODUCTION

A. Virtual reality

Virtual reality [1] is a simulated experience that can be similar to or completely different from the real world. This experience is created with of technical means and exists in numerical form. The created virtual effects are projected onto human consciousness allowing to experience sensations close to reality [2, 3]. VR simulators, both gaming and training ones, are widely applied in various engineering fields.

B. Virtual reality (VR) glasses

VR glasses [4] (Fig. 1) are comprised of hard plastic shell with straps. Inside the shell lenses are placed with one or two screens behind. In our case one screen for each eye is provided. Imaging for both eyes can be either the same or different in the case of 3D content. In order to orient in virtual world, the device responses to user head rotations are required. This is aided by gyroscopes and accelerometers. The acquired data are transferred to PC via USB [5] or HDMI [6] cables.

Revised Manuscript Received on November 30, 2019.
* Correspondence Author
Sergei Sergeevich Smirnov*, Complex Systems LLC, Tver, Russia
Dmitrii Aleksandrovich Sytnik, Complex Systems LLC, Tver, Russia
Nikolai Valerievich Bradis, Complex Systems LLC, Tver, Russia
Andrei Ivanovich Kardakov, Intelligent Solutions LLC, Tver, Russia

Fig. 1: VR glasses

C. StarBlade VR simulator

StarBlade [7] (Fig. 2) is comprised of a platform with servo motors. This platform is positioned horizontally and can rotate around vertical axis. The cabin with user's seat is located on the platform, it can tilt backward and forward by certain angle. It is assumed that a user occupies the seat, puts on VR glasses, and tries VR experience. Therefore, the user is able not only to tilt backward and forward, to rotate around vertical axis, but also to rotate head which adds more sensations upon trying VR experience.

Fig. 2: StarBlade VR simulator

Cabin position can be estimated by stationary rotation sensors. However, their mounting is accompanied by engineering problems, which increases the simulator cost.

Thus, in this work it is proposed to apply inertial sensors instead of expensive rotation sensors. The modules for installation of inertial sensors are relatively inexpensive, their sizes are moderate: about 2×3 cm. They can by placed in any point of the cabin. The sensors can be either wireless (BluetoothLE [8]) or wired (USB, HDMI).

D. VirtuSphere VR simulator

The sphere (Fig. 3) is comprised of three plastic segments. The segments are connected by special locks. One segment is detachable. It is used for user entry into the sphere. The sphere diameter is 3 m. The sphere is positioned on special frame with rollers. The rollers are mounted on mechanisms which can rotate around their axes. The rotating mechanisms are positioned on the frame in several points providing free rotation of the sphere. The sphere can rotate in all directions including its vertical axis.

Fig. 3: VirtuSphere VR simulator.

The sphere is used for VR systems. Visualization is provided by VR glasses, the sphere is intended for moving across virtual environment. A user enters the sphere and starts to move across the virtual environment performing certain actions. In order to provide current picture in virtual space, it is required to know direction vector and linear speed of the user in the sphere.

E. Formulation of the purpose

In order to integrate the aforementioned facilities into VR environment, it is required to know the position of simulator in space. This is aided by inertial sensors, namely: gyroscope, accelerometer, and magnetometer. Application of each sensor separately cannot track position of simulator in space. For instance, accelerometer cannot track rotations around axes. Gyroscope can track rotations around axes, however, error accumulates related with integration of angular speeds. Hence, it is insufficient to determine orientation by means of only gyroscope. Magnetometer can be used as compass, however, it is very difficult to track rotations using it. Generally, the modern systems of inertial tracking are equipped with several sensors. We will use three-axis gyroscope and three-axis accelerometer for tracking of VR glasses and Sphere simulator. Three-axis gyroscope, accelerometer, and magnetometer will be used for StarBlade simulator. Position in space will be determined by the advanced method: Madgwick filter.

II. METHODS

A. Tools

ICM-20602 inertial sensors [9] integrated into wireless module based on STM32 [10] microcontroller with BluetoothLE 4.0 interface [8] and MPU 9350 [11] sensors based on STM32 [10] microcontroller with HID interface [12] connected to PC via USB port were used as tools in this work. These sensors are equipped with three-axis gyroscope, three accelerometers for each axis and three-axis magnetometer.

B. Inertial sensors data digitization

Generally, all digital sensors convert analog signal into digital signal. This is aided by analog-to-digital converters (ADC) [13]. In its turn, ADC is characterized by bit depth which defines the number of discrete values provided by converter at output. In binary ADC, the resolution is measured in bits. For instance, 16-bit ADC is capable to provide 2^{16} = 65,536 discrete values. In order to convert abstract data into obvious physical quantities, it is required to convert ADC units into respective physical quantities. Let us exemplify this conversion by accelerometer. Let the accelerometer range is k = ±2g (g=9.80666 m/s^2: acceleration of gravity) and ADC depth is 16 bit (*depthADC* = 16), then, ADC units are converted into acceleration units by Eq. (1):

$$val = \frac{k \times raw}{depthADC} (1),$$

where *val* is the value in physical quantities (in this case: m/s^2), *k* is the sensor range (the value in physical quantities, in this case: m/s^2), *raw* is the sensor readings in ADC units, *depthADC* is the ADC bit depth.

Hence, the ADC units are converted into physical quantitates.

C. Calibration of inertial sensors

Prior to application of the sensor data, it should be calibrated. With this aim, a sensor is positioned appropriately and its readings are acquired. Then these data are compared with reference data, the type of calibration curve is determined. For instance, gyroscope in rest state should yield zero. Therefore, for gyroscope it is possible to apply only zero shift which is determined upon calibration. In addition, it is important to account for non-orthogonality of sensor axes upon calibration. Magnetometer requires for more complex calibration, and it should be performed in actual place of sensor application because the lines of earth magnetic field intensity are not always parallel to its surface.

D. Flow Chart

Figure 4 illustrates general flowchart of data preparation for computations.

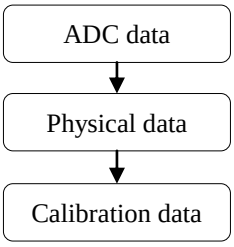

Fig. 4: Data preparation for computations

E. Gyroscope data digitization

Operation modes of gyroscope of ICM-20602 and MPU 9350 sensors are summarized in Table 1. It can be seen that the operation modes of gyroscopes are nearly similar.

The gyroscope of ICM-20602 sensor has five operation modes, and the gyroscope of MPU 9350 sensor – four operation modes. The units of both sensors are the same: degree/second. Gyroscopes determine angular speed with regard to rotation axis. These sensors are based on three axes with regard to which the angular speed is calculated. These are Ox, Oy and Oz axes.

Table 1: Gyroscope modes for ICM-20602 and MPU 9350 sensors

Mode No.	ICM-20602 range	MPU 9350 range	Units
1	±200	±250	degree/second (dps)
2	±400	±500	degree/second (dps)
3	±500	±1,000	degree/second (dps)
4	±1,000	±2,000	degree/second (dps)
5	±2,000		degree/second (dps)

In order to determine orientation of inertial sensor, it is required to integrate readings over the time with respect to each axis. Upon integration of sensor readings, an error accumulates, which increases measurement error in time. This accumulated error is referred to as gyroscope zero shift. A factor is introduced to eliminate this error. This coefficient is selected individually for each sensor upon calibration.

ICM-20602 sensor is equipped with 16-bit ADC, hence, the digitization range is 2^{16} digits. For positive values, the minimum is 0, the maximum is 65,535. For numbers with sign, the minimum is –32,768, the maximum is 32,767. ADC units can be converted into degree/second using Eq. (1). For the range of ±2,000 degree/second (dps), Eq. (1) is as follows:

$$\omega = \frac{2{,}000 \times raw}{32{,}768}, (2)$$

where ω is the angular speed (degree/second), *raw* is the sensor readings.

Therefore, we obtain Eqs. (3), (4), (5) for conversion of ADC units into degree/second for the Ox, Oy and Oz axes.

$$\omega_x = \frac{2{,}000 \times raw_x}{32{,}768} (3)$$

$$\omega_y = \frac{2{,}000 \times raw_y}{32{,}768} (4)$$

$$\omega_z = \frac{2{,}000 \times raw_z}{32{,}768} (5),$$

where ω_x is the angular speed with regard to Ox axis , ω_y is the angular speed with regard to Oy axis, ω_z is the angular speed with regard to Oz axis.

In order to convert into radian/second, it is required to multiply the previous conversion by π and to divide by 180°.

F. Accelerometer data digitization

Operation modes of accelerometer of ICM-20602 and MPU 9350 sensors are summarized in Table 2. It can be seen that the operation modes of accelerometers are the same. The accelerometer of ICM-20602 sensor and the accelerometer of MPU 9350 sensor have four operation modes. The units of measurement of both sensors are the same: m/s^2.

Table 2: Accelerometer modes of ICM-20602 and MPU 9350 sensors

Mode No.	ICM-20602 range	MPU 9350 range	Units
1	±2	±2	m/s^2
2	±4	±4	m/s^2
3	±8	±8	m/s^2
4	±16	±16	m/s^2

Accelerometers determine projection of each axis onto gravity vector. These sensors are based on three axes with regard to which the projections are calculated. In order to determine orientation of inertial sensor in space, it is required to know the projections of axes onto *g* vector. Unfortunately, using accelerometer readings, it is impossible to track certain rotations, for instance, rotation around Oz axis.

ICM-20602 sensor is equipped with 16-bit ADC, therefore, the digitization range is 2^{16} digits. For positive values, the minimum is 0, the maximum is 65,535. For numbers with sign, the minimum is –32,768, the maximum is 32,767. ADC units can be converted into degree/second using Eq. (1). For the range of ±2g, Eq. (1) is as follows :

$$a = \frac{2 \times raw}{32{,}768} (6)$$

where *a* is the acceleration of gravity (9.80666 m/s^2), *raw* is the sensor readings.

Therefore, we obtain Eqs. (7), (8),(9) for conversion of ADC units into m/s^2 for each axis:

$$a_x = \frac{2 \times raw_x}{32{,}768}, (7)$$

$$a_y = \frac{2 \times raw_y}{32{,}768}, (8)$$

$$a_z = \frac{2 \times raw_z}{32{,}768}, (9)$$

where a_x is the projection of gravity vector onto Ox axis, a_y is the projection of gravity vector onto Oy axis, a_z is the projection of gravity vector onto Oz axis

G. Magnetometer data digitization

ICM-20602 sensor is not equipped with magnetometer, and we will discuss MPU 9350 sensor. The measurement range of magnetic field is ±4,800 μT (microtesla). MPU 9350 sensor is equipped with 14-bit ADC, thus, the digitization range is 2^{14} digits. For positive values, the minimum is 0 and the maximum is 8,192. For numbers with sign, the minimum is –8,192 and the maximum is 8,191. ADC units can be converted into μT using Eq. (1). For the range of ±4,800(μT), Eq. (1) is as follows :

$$H = \frac{4{,}800 \times raw}{8{,}191} (10)$$

where *H* is the vector projection of earth magnetic field intensity (μT), *raw* is the sensor readings.

Therefore, we obtain Eqs. (11), (12,(13) for conversion of ADC units into μT for each axis:

$$H_x = \frac{4{,}800 \times raw_x}{8{,}192} (11)$$

$$H_y = \frac{4{,}800 \times raw_y}{8{,}192} (12)$$

$$H_z = \frac{4{,}800 \times raw_z}{8{,}192} (13)$$

where H_x is the projection of earth magnetic field intensity onto Ox axis, H_y is the projection of earth magnetic field intensity onto Oy axis, H_z is the projection of earth magnetic field intensity onto Oz axis.

Since the vector of magnetic field is not always parallel to earth surface, it is required to perform additional sensor calibration [14] .

H. Description of 3D rotations, quaternions

In this work, the coordinate rotation is described using the notion of quaternion. A quaternion [15, 16] (Latin: *quaterni*, by four) is system of hypercomplex numbers making a 4-dimensional vector space over the real numbers. Quaternions were first described by William Rowan Hamilton in 1843. Using quaternions, it is convenient to describe rotations of bodies in space. As applied to rotation, quaternion is a three-dimensional vector defined by rotation axis and scalar, which determines angle of rotation around preset axis.

I. Madgwick filter

Data from inertial sensors in dynamic mode (when sensors rotate or move) are noise polluted and require for additional filtration. Such noises are generally suppressed by complementary filters: Kalman filter [17], Madgwick filter [18], and others. The advantage of Madgwick filter in comparison with others in our case is better smoothing of noises as well as lower computation load of processor, which facilitates its implementation using microcontrollers. In addition, the filter uses the method of gradient descent for correction of accelerometer readings by gyroscope readings, thus providing better sensor orientation in space even at high rotation speeds. This filter is based on application of quaternions [16]. According to the authors, this filter provides better results in comparison with complementary filters based on Kalman method at least in terms of productivity and results. This work was successfully presented in the Bristol university. This method is implemented in two algorithms. The first algorithm is based on data of gyroscope and accelerometer sensors. In the second algorithm, gyroscope and accelerometer are supplemented by magnetometer. The first algorithm has one adjustable parameter, the second one has two parameters. Flowchart of coordinate rotation using Madgwick filter is illustrated in Fig. 5. Data from inertial sensors are converted into physical quantities, the data are calibrated. Then, the prepared data are transferred to filter. At the filter output, quaternion is obtained which is then inverted. Coordinate rotation is based on the obtained quaternion.

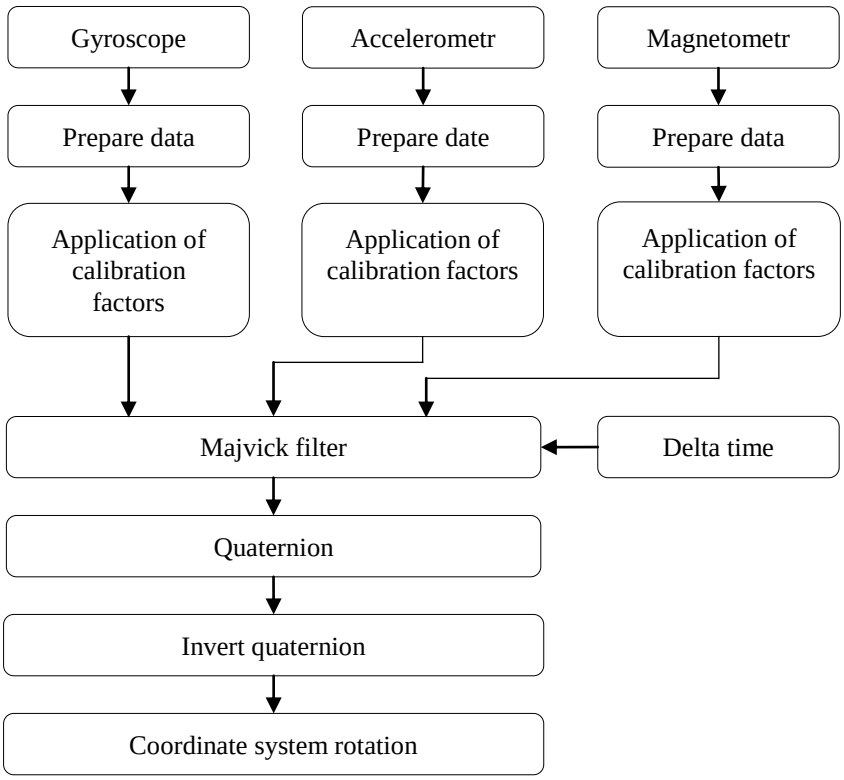

Fig. 5: Coordinate rotation flowchart

J. Tracking of VR glasses

Tracking of VR glasses is based on gyroscope sensor and accelerometer sensor. In addition, the time of arrival of each data packet was taken into account because the interval of data arrival from sensors is not constant and varies in the range from 10 to 30 ms. Accounting for the interval of data arrival increases integration accuracy of angular speeds. Moreover, while tuning the Madgwick filter, the factor is selected responsible for gyroscope zero shift.

The filter application results in quaternion which describes sensor rotation with regard to global coordinates. Since a user observes global coordinates (scene) using glasses, that is, in sensor coordinates, then, we should transfer from global coordinates to sensor coordinates. With this aim, the obtained quaternion should be inverted. Then, this quaternion should be applied for scene rotation. In such manner the effect of participation in virtual experience is achieved.

K. StarBlade tracking

Tracking of StarBlade simulator is performed similarly to tracking of VR glasses with the exception that the Madgwick filter uses magnetometer readings. Thus, obtained quaternion is also inverted. For clear representation of StarBlade cabin rotation, it is necessary to present quaternion in the form of plain angles. Therefore, rotations are presented in the form of Euler angles α, β (yawing, pitching). That is, the quaternion is converted into angles. Conversion of quaternions into Euler angles is described in details in [16].

L. VirtuSphere tracking

Tracking of VirtuSphere simulator is similar to that in VR glasses.

The distance during sphere rotation is determined by vectors. The vectors determining previous and current position of Oz axis are used. Then the angle γ between these vectors is calculated. These angles are added as follows:

$$\alpha = \sum \gamma_i, \ (14)$$

where α is the cumulative rotation angle (radian), γ is the angle between vectors.

The covered distance is calculated as follows:

$$l = \frac{D * \alpha}{2}, (15)$$

where *l* is the distance in meters, *D* is the sphere diameter in meters, *α* is the cumulative rotation angle in radians.

The linear speed of sphere rotation in horizontal plane is determined by vector stack. The instant linear speed is calculated by sensor sampling frequency and sphere diameter. Measurement interval is calculated as follows:

$$t = \frac{N}{f}, (16)$$

where *N* is the number of vectors in stack, *f* is the sensor sampling frequency, *t* is the measurement interval.

The distance covered per the time *t* is calculated as follows:

$$l = \frac{D * \alpha_t}{2}, (17)$$

where *l* is the distance in meters, *D* is the sphere diameter in meters, α_t is the cumulative rotation angle in radians per time *t*.

Therefore, we have:

$$v = \frac{l}{t}, (18)$$

where *v* is the linear speed of sphere rotation, *l* is the distance per time *t*.

As a consequence of filter operation, the sensor axis coordinates can be slightly shifted, hence, the minimum detected speed is limited by 0.3 m/s.

Direction of sphere rotation is determined by tracking Oz axis sensor and comparing with previous position. Two cases can be highlighted. The first case: when the new position of Oz` axis differs significantly from previous position Oz. The second case: when the position of Oz axis is not varied. In the first case, it is possible to state that the sphere rotates in any direction. The second case is comprised of two variants.

The first variant is the simplest: the sphere is at rest. The second variant is as follows: the sphere rotates around Oz axis. In order to differentiate these two variants, it is necessary to detect position of Ox axis or Oy axis. If these axes are at rest (their coordinates remain the same), then it is possible to state that the sphere is at rest, otherwise, the sphere rotates around Oz axis. In the first case, the coordinates of Oz axis of simulator vary continuously. Direction of sphere rotation is determined by spherical coordinates. This is aided by conversion of coordinates of Oz` vector from Cartesian to spherical coordinates. Current direction of sphere rotation is determined using the angle φ as follows:

$$\varphi_m = \varphi, (19)$$

where φ_m is the angle determining direction of sphere rotation, φ is the polar angle of Oz` vector.

In the case when sphere rotates around Oz axis, the main direction, predefined by the angle φ_m, is supplemented by the angle φ_r. The angle φ_r is based on initial position of Ox`axis, and we calculate the polar angle $\varphi_{old}$ for this position. For all other positions of Ox` vector, we calculate the polar angle φ_{cur}. The coordinates of vector Ox` are recorded up to termination of rotation around Oz axis. Cumulative angle of direction is determined as follows:

$$\varphi_r = \varphi_{old} - \varphi_{cur}, (20)$$

where φ_r is the sphere rotation angle around Oz axis, φ_{old} is the initial rotation angle, φ_{cur} is the current rotation angle.

Direction of sphere rotation is determined as follows:

$$\varphi = \varphi_m - \varphi_r, (21)$$

where φ is the angle determining direction of sphere rotation, φ_m is the angle determining direction of sphere rotation, φ_r is the angle of sphere rotation around Oz axis.

Algorithm of determination of linear speed and direction of sphere rotation is illustrated in Fig. 6.

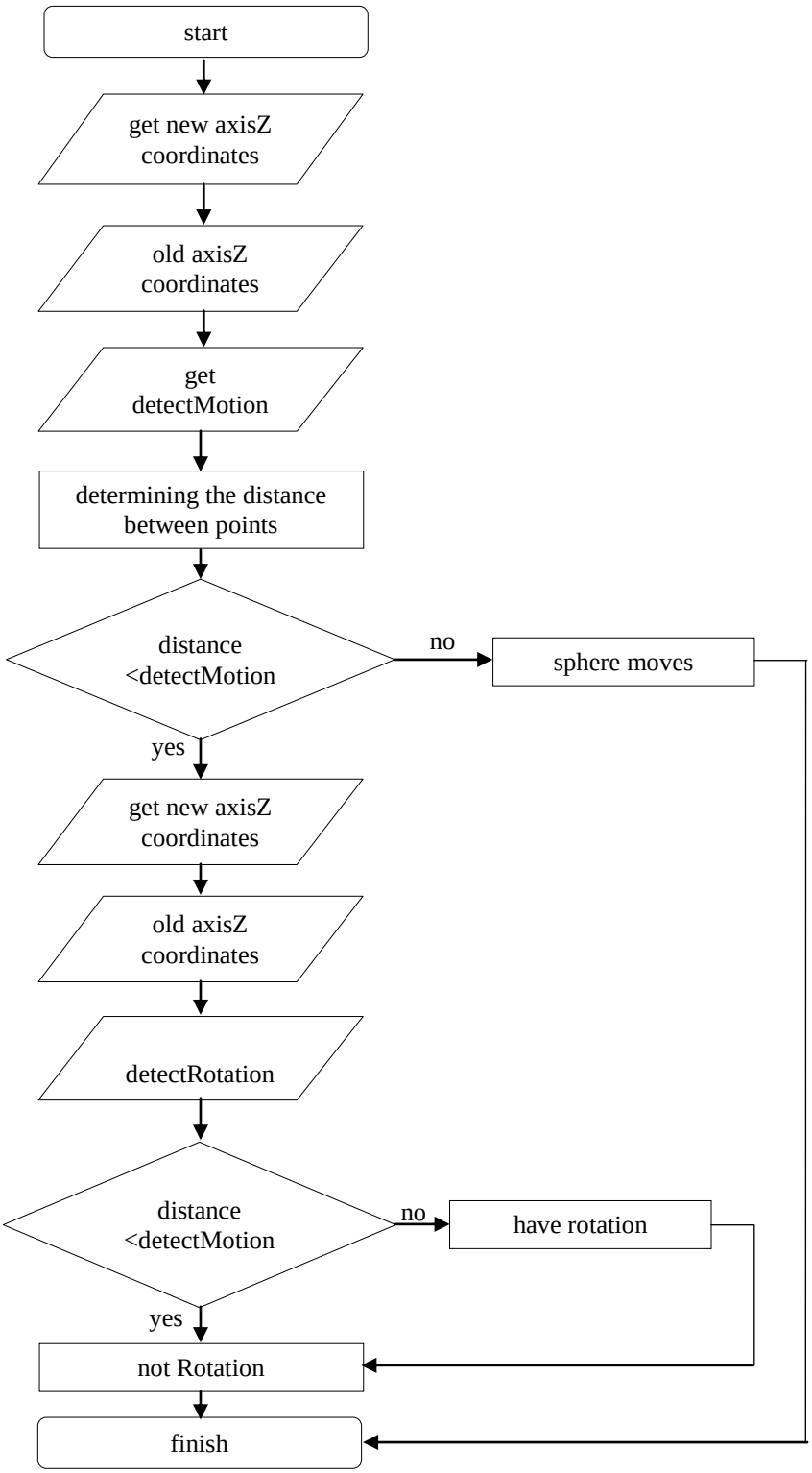

Fig. 6: Sphere tracking algorithms.

III. RESULTS AND DISCUSSION

Tracking algorithm for VR glasses was developed and implemented. The developed algorithm was successfully applied for StarBlade VR simulator. In the case of VirtuSphere VR simulator, the algorithm was used for computation of motion parameters, namely: distance, speed, direction (Fig. 7).

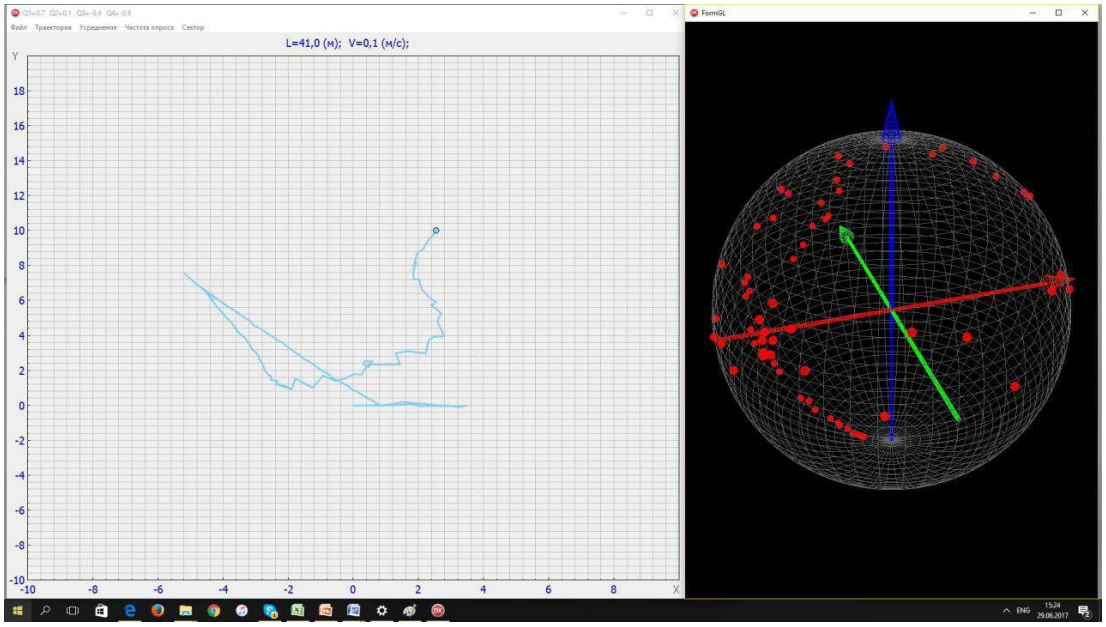

Fig. 7: User motion path in sphere

Application of this approach to tracking was successful. However, in the course of time the virtual position was somewhat displaced. This is related with gyroscope zero drift. Therefore, VR glasses should be subsequently supplemented with video tracking (6DOF).

In the case of StarBlade VR simulator, the results are mainly satisfactory due to application of magnetometer during overall time of simulator operator: the displacement of initial cabin position was in the range of one degree.

Only one module with sensors was used for sphere tracking. When the module is located under user's feet, there occurs significant bounce of sensor data. This bounce can be eliminated by installation of additional module at opposite side of the sphere. Figure 7 illustrates test results of sphere tracking.

IV. CONCLUSION

Therefore, it is possible to state that 3D tracking based on Madgwick filter provides good results. This filter can be implemented not only in desktop computers but also in tracking microcontrollers.

ACKNOWLEDGMENT

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Project No. 14.576.21.0102, Unique identifier: RFMEFI57617X0102).

REFERENCES

1. Virtual reality. Wikipedia, the free encyclopedia. Available at: https://en.wikipedia.org/wiki/Virtual_reality
2. A.S. Nabiev, L.T. Nurkusheva, K.K. Suleimenova, G.K. Sadvokasova, Z.A. Imanbaeva, “Virtual Reconstruction of Historical Architectural Monuments: Methods and Technologies”, International Journal of Innovative Technology and Exploring Engineering, 8(10), 2019, pp. 3880-3887.
3. E.A. Kirillova, A.V. Pavlyuk, I.A. Mikhaylova, T.E. Zulfugarzade, S.S. Zenin, “Bitcoin, Lifecoin, Namecoin: The Legal Nature of Virtual Currency”, Journal of Advanced Research in Law and Economics, 9(1), 2018, pp. 119-126.
4. Total Vision. Available at: http://total-vision.ru/
5. USB. Wikipedia, the free encyclopedia. Available at: https://en.wikipedia.org/wiki/USB
6. HDMI. Wikipedia, the free encyclopedia. Available at: https://en.wikipedia.org/wiki/HDMI
7. StarBlade: virtual reality simulator. Total Interactive. Available at: https://total-interactive.com/content/products/star_blade/
8. Bluetooth Low Energy. Wikipedia, the free encyclopedia. Available at: https://en.wikipedia.org/wiki/Bluetooth_Low_Energy
9. ICM-20602 High Performance 6-axis MEMS MotionTracking Device. TDK InvenSense. Available at: https://www.invensense.com/products/motion-tracking/6-axis/icm-20602/
10. STM32 32-bit Arm Cortex MCUs. STMicroelectronics. Available at: https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html
11. Introduction Of The World’s Lowest Power 9-Axis MotionTracking Device. Electronic Specifier. Available at: https://www.electronicspecifier.com/sensors/mpu-9350-invensense-worlds-lowest-power-9-axis-motiontracking-device
12. USB human interface device class. Wikipedia, the free encyclopedia. Available at: https://en.wikipedia.org/wiki/USB_human_interface_device_class
13. Analog to digital converter. Wikipedia, the free encyclopedia. Available at: https://en.wikipedia.org/wiki/Analog-to-digital_converter
14. O. Evsegneev, “Kalibrovka magnitometra. Elektronnyi kompas” [Calibration of magnetometer. Electrnoic compass]. Robotclass. Available at: https://robotclass.ru/articles/magnetometer-and-compass
15. Quaternion. Wikipedia, the free encyclopedia. Available at: https://ru.wikipedia.org/wiki/Кватернион
16. V.N. Gordeev, “Kvaterniony i bikvaterniony s prilozheniyami v geometrii i mekhanike” [Quaternions and biquaternions with applications in geometry and mechanics]. Kiev: Stal', 2016.
17. E. Foxlin, “Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter”. In Proc. *Virtual Reality Annual International Symposium the IEEE*, 267, 1996, pp. 185–194.
18. S.O.H. Madgwick, “An efficient orientation filter for inertial and inertial/magnetic sensor arrays”, Report x-io and University of Bristol (UK), 25, 2010, pp. 113–118.