IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 April 2023, accepted 28 May 2023, date of publication 1 June 2023, date of current version 8 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3281970

RESEARCH ARTICLE

DO IONet: 9-Axis IMU-Based 6-DOF Odometry Framework Using Neural Network for Direct Orientation Estimation

HONG-IL SEO[1], JU-WON BAE[1], WON-YEOL KIM[2], AND DONG-HOAN SEO[3]

[1]Department of Electronics and Electrical Engineering, Interdisciplinary Major of Maritime AI Convergence, Korea Maritime and Ocean University, Busan 49112, South Korea
[2]Department of Artificial Intelligence Engineering, Chosun University, Gwangju 61452, South Korea
[3]Division of Electronics and Electrical Information Engineering, Interdisciplinary Major of Maritime AI Convergence, Korea Maritime and Ocean University, Busan 49112, South Korea

Corresponding author: Dong-Hoan Seo (dhseo@kmou.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government [Ministry of Science and ICT (MSIT)] under Grant NRF-2021R1A2C1014024.

ABSTRACT With the development of mobile devices, low-cost inertial measurement unit (IMU)-based research on inertial odometry for indoor localization is being actively conducted. Inertial odometry estimates the amount of change in position and orientation relative to the initial value based on the data measured by the IMU and creates a trajectory via a multi-integration process. However, existing approaches primarily focus on estimating the position in a two-dimensional (2D) plane. Additionally, drift errors occur because the position is estimated by integrating the position change and the orientation change. Herein, we propose a novel six-degree of freedom inertial odometry framework that directly estimates the orientation to minimize drift errors. The proposed framework is a direct-orientation inertial odometry network (DO IONet) that outputs the velocity and orientation of the IMU by using linear acceleration, gyro data, gravitational acceleration, and geomagnetic data as inputs to structurally eliminate drift errors that occur during orientation calculations. DO IONet is composed of a convolutional neural network-based encoder to extract features from inertial data and decoder to extract sequential features. Structurally, it does not require initial values and has no cumulative error despite estimating orientation over tens of seconds.

INDEX TERMS 6 degrees of freedom, inertial measurement units, inertial odometry, neural network.

I. INTRODUCTION

Odometry has a wide range of applications in fields requiring self-localization, such as human motion monitoring (healthcare, athletics, gaming) and navigation systems (drone, vehicle, vessel). Odometry aims to realize self-localization by allocating a pose, which contains a position and an orientation over time, using data collected from various sensors [1], [2]. Depending on the type of sensor, the odometry methods are classified into four types: visual odometry (VO) using a camera, radar odometry (RO), lidar odometry (LO), and inertial odometry (IO) using an inertial measurement unit (IMU) [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Halil Ersin Soken.

VO is a technique used to estimate changes in the pose of a platform by analyzing the variations using a sequence of images collected by the camera, which can be an RGB-depth, stereo, or monocular camera [3], [4], [5]. Although VO techniques exhibit good localization performance, fundamental problems are caused by using images, such as scale ambiguity and computational complexity. RO is an approach for estimating the relative motion of a platform by analyzing scans obtained from a radar sensor using radio waves to determine the range, angle, and velocity of the surrounding objects [6], [7]. In particular, RO has attracted attention owing to its robustness in poor weather conditions. LO estimates a localization method by analyzing laser speckle patterns reflected from surrounding objects [8], [9]. The laser is capable of

providing a three-dimensional (3D) scan depending on the system design. Therefore, it is possible to collect precise information regarding surrounding objects. However, it is difficult to apply RO and LO to resource-limited platforms, such as smartphones and tablets, because the aforementioned methods require high computational power owing to the extremely large amount of input data that is necessary for estimating the location of the platform using the surrounding information. IO is an approach for estimating the pose of a platform relative to a reference point by using an IMU sensor. An IMU sensor is a micro-electro-mechanical system (MEMS)-based device that measures three-axis acceleration, three-axis gyro data, and three-axis geo-magnetic data [10], [11], [12]. The volume of data measured every hour in the IMU includes 3D motion information and is significantly smaller than the volume of data stored in images and 2D maps. Therefore, IO is an essential technology for the operation of real-time positioning systems.

Inertial odometry continuously estimates the amount of relative pose change using the data measured by the combination of inertial sensors (accelerometer, gyroscope, and geomagnetic). This approach is used in environments where it is difficult to use well-established infrastructure-based methods, such as the Global Positioning System (GPS), Wi-Fi, and Bluetooth Low Energy (BLE), and can measure the precise position within 1m [13], [14], [15]. However, these techniques involve two inherent problems: drift error and initial pose setting. Drift error is a phenomenon in which the estimated pose continuously diverges because noise is accumulated and integrated in the process of generating the trajectory by using the angular and linear velocity obtained from the initial pose. This problem can be resolved by lowering the dimension of the trajectory to be generated or by setting a reference point for initialization; however, these are not fundamental solutions. The initial pose setting problem is that the initial orientation of the mobile device must be specified to generate a trajectory in the reference coordinate system. For example, if you generate a 3D trajectory that travels straight on flat ground without a known orientation, the angle of the travelling path is unknown. Therefore, it is essential to solve the above two problems to significantly improve the performance of IO.

Representative IO approaches include strapdown inertial navigation system (SINS) and pedestrian dead reckoning (PDR). SINS realizes 3D localization by integrating the amount of pose change [16], [17], [18], [19], [20]. The current orientation is calculated by continuous input of the gyroscope value, which is the angular velocity. The position is calculated by integrating the linear acceleration with the gravitational acceleration removed from the acceleration rotated by the current orientation in the reference coordinate system. Although this approach is the most intuitive and fundamental, it is difficult to generate a long-duration trajectory owing to the continuous drift error. PDR outputs the 2D position of a pedestrian using step, string length, and heading calculated via each algorithm [20], [21], [22], [23]. This system is almost free from drift error and exhibits excellent trajectory estimation performance because the orientation is continuously initialized via zero velocity update (ZUPT). However, this method requires precise sensor calibration and can only be used for 2D odometry [24].

Recently, an inertial odometry network (IONet), which is an IO using a low-cost IMU based on deep neural network, has been proposed [25]. Most IONets output the changes in pose by inputting the sequential data obtained from the IMU sensor into a recurrent neural network. Finally, the integrated value of change generates a trajectory. The outcome of IONet is similar to the traditional method; however, it reduces the drift error by inputting all sequential data. However, similar to SINS, this method requires the reference value, and the correct orientation of the sensor cannot be confirmed owing to the drift error and structural problems with the continuous integration process as the estimation time increases.

In this study, to constrain the orientation drift error of the inertial system, we propose a novel framework—the direct-orientation inertial odometry network (DO IO-Net)—that estimates the position change and achieves continuous orientation using end-to-end learning. Compared with the method of integrating inertial data, the proposed approach does not integrate the angular velocity and orientation output because it uses the gravitational acceleration and geomagnetic data, including the information regarding the inclination angle and the azimuth, respectively. Therefore, the orientation drift error does not occur. In addition, the proposed model does not require the initial value of orientation in the reference coordinate system and can continuously estimate the trajectory. The DO IONet, which consists of two layers of a convolutional neural network (CNN) and two layers of bidirectional long short-term memory (LSTM), outputs the position change and orientation using the raw data of a nine-axis IMU.

The major contributions of this work are summarized as follows:

1. We propose a novel IONet framework, named DO IONet, that estimates the position changes and orientation using the data from a nine-axis IMU sensor.

2. The proposed framework directly estimates the orientation to reduce drift errors that occur during trajectory generation.

3. The proposed method can be easily applied to other odometry within a reference coordinate system without a reference orientation value.

II. RELATED WORK

A. 3-DOF INERTIAL ODOMETRY NETWORK

In the field of odometry, the degree of freedom (DOF) is the number of axes used to represent the position. In a 3-DOF IO, the relative position change in the 2D plane is expressed using the distance and azimuth (yaw angle). 3-DOF IONet refers to inertial odometry using a neural network. The results of 3-DOF odometry are integrated to generate a 2D trajectory. The initial learning-based approach used a support vector machine (SVM), similar to PDR, to classify the

location of the IMU [26]. Subsequently, the position change and azimuth were estimated by using support vector regression (SVR) according to the IMU location. IONet has developed an Oxford Inertial Odometry Dataset (OxIOD) containing inertial data (linear acceleration, gravitational acceleration, gyro data, and geomagnetic data) and is based on LSTM for estimating 2D trajectory of the sensor unit [25]. These methods can estimate the amount of general position change using raw inertial data without requiring a periodic pattern. The author of [27] improved the positioning performance by directly calculating the position using raw sensor data from a calibrated 9-axis IMU and 1D CNN-based velocity estimation model. The author mentioned in [28] proposed a new loss function based on the analysis of the IMU kinematic equation to improve the performance variation of each device. However, these methods can only be applied to 2D odometry, which involves position sensing using the attached sensors and additional assumptions similar to PDR.

B. 6-DOF INERTIAL ODOMETRY NETWORK

6-DOF IO represents the relative pose change in 3D space using the position (x-axis, y-axis, z-axis) and orientation (roll angle, pitch angle, and yaw angle) of the object. Because transformation from the body frame to the reference coordinate system is required for each sequence, considerable drift errors occur compared with 3-DOF inertial odometry. The initial 6-DOF IONet estimated the 3D pose change of the sensor unit using linear acceleration and a gyro data [29]. The result of the model generated the orientation and 3D trajectory via integration. However, it is difficult to estimate long-term trajectory and orientation using this method because it additionally estimates the position change and orientation, which was not considered in the traditional inertial odometry framework. Extended IONet additionally inputted gravitational acceleration and geomagnetic data that can be directly used for the orientation calculation and used a model for estimating orientation to only improve the accuracy [30]. As a result, the performance of the trajectory generation was significantly improved, and the drift error was decreased due to the reduction in the orientation error. However, this method diverges as the estimation time increases, and it cannot be corrected. One solution to improve orientation accuracy is to utilize pre-processed inertial data as input to a neural network to estimate position and orientation change, and use a 2D lidar to correct orientation [31]. However, this method only targets limited systems and was tested using simulated data. Despite numerous attempts to improve orientation quality, it is difficult to eliminate errors that arise during the computation of orientation. Therefore, a novel inertial odometry framework is required to minimize the cumulative error by directly estimating the orientation.

III. DIRECT-ORIENTATION INERTIAL ODOMETRY FRAMEWORK

We propose a novel 6-DOF inertial odometry framework, DO IONet, to eliminate the drift in orientation estimations.

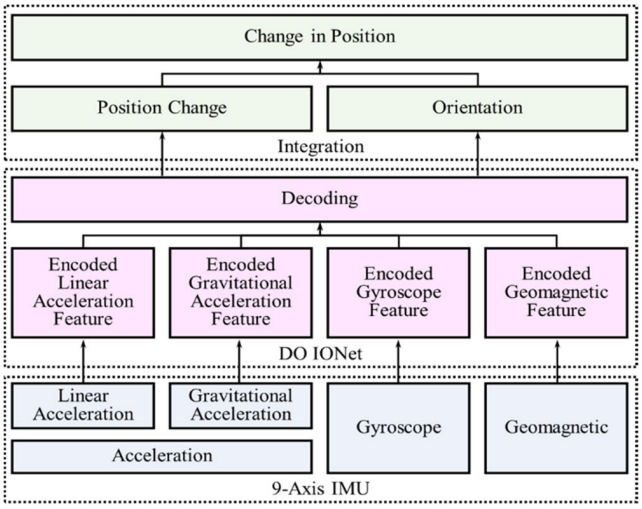

FIGURE 1. Overview of the proposed framework.

The proposed framework is depicted in Fig. 1, and it comprises a 9-axis IMU representing input data, DO IONet processing data, and trajectory generation. DO IONet is composed of an encoding step separated for each data and a decoding step in which the features are concatenated to output the position change and orientation. The trajectory is generated by integrating the values output from the neural network. The proposed framework estimates the orientation directly; therefore, it does not structurally cause orientation drift and does not require an initial value in the reference coordinate system.

A. NETWORK ARCHITECTURE

Fig. 2 illustrates the architecture of the proposed network that inputs the time-series data and outputs the position change and orientation. The input IMU data are linear acceleration a, angular velocity w, gravitational acceleration g, and geomagnetic value m. One set of input data comprises 200 frames of consecutive IMU data every 10 strides, and the position difference Δp with sensor orientation is removed between the 95th and 105th data points and the 95th orientation q. The estimated orientation uses quaternions for the convenience of learning instead of Euler angles with circular features. The structure of the proposed network consists of 1D CNN layers, Bi-LSTM layers, and fully connected layers. Two 1D convolutional layers encode features extracted from the input IMU data, respectively.

Subsequently, the features output from the CNN are compressed using Max pooling to reduce the complexity of the features and prevent overfitting. Each compressed feature is concatenated and input to the Bi-LSTM layer used for sequential data processing. The Bi-LSTM layer extracts overall patterns by endowing temporal dependencies into the features. Specifically, it acts as a decoder that connects each time step's feature to regress to a relative position. In addition, the bidirectional model mitigates the vanishing gradient problem

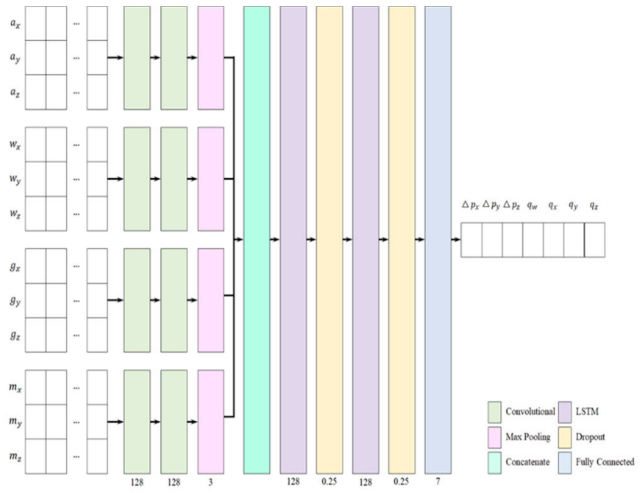

FIGURE 2. Structure of DO IONet.

and enables the model to learn more patterns compared to a conventional recurrent neural network model. Finally, Δp and q are estimated through the fully connected layer, where Δp represents the 3D translation vector in the body-frame, and q denotes the orientation in the reference coordinate system.

B. 6-DOF RELATIVE POSE REPRESENTATION

In the neural network-based inertial odometry approach, the unit quaternion is used because the network cannot learn when using Euler angles. Similar to [30], we used a 3D translation vector, previous position, current position, and previous orientation in the reference coordinate system to represent pose changes, as expressed in (1),

$$p_t = p_{t-1} + R\left(q_{t-1}\right)\Delta p, \tag{1}$$

where p_t denotes the current position calculated between the previous position p_{t-1} and the rotation matrix $R(q)$ for the previous orientation. Accordingly, DO IONet structurally reduces the drift compared with traditional inertial odometry because the integration process of orientation is omitted.

C. LOSS FUNCTION

6-DOF IONet designs loss functions for both position and orientation. However, as position and orientation have different scales, the homoscedastic uncertainty of each task needs to be considered. Therefore, we used multi-task loss to learn various quantities on different scales [32],

$$L_{total} = \sum_{i=1}^{n} e^{-log\sigma_i^2} L_i + log\sigma_i^2, \tag{2}$$

where σ_i and L_i denote the variance and loss function of the i^{th}, respectively. As explained in [30], the exponential mapping unconstraint the scalar values, and the log variance increases the stability of the network.

There are two detailed loss functions, as expressed in (3),

$$[L_1, L_2] = \left[L_{DPMAE}, L_{QME}\right]. \tag{3}$$

L_{DPMAE} represents the delta position mean absolute error, and L_{QME} denotes the quaternion multiplicative error,

$$L_{DPMAE} = \left\|\Delta p - \Delta\hat{p}\right\|_1, \tag{4}$$

$$L_{QME} = 2 * \|imag(\hat{q} \otimes q^*)\|_1. \tag{5}$$

L_{DPMAE} and L_{QME} represent the loss functions for the velocity of sensor unit without orientation and the orientation in the reference coordinate system, respectively. The $imag(q)$ returns a complex number as a real number, and $\otimes$ symbolizes the Hamilton product operator for the sum of two angles. q^* indicates the conjugate complex number of q.

IV. EXPERIMENTS

A. DATASET

For our experiment, we utilized the highly precise OxIOD, which provides four types of inertial data: linear acceleration, gyro data, gravitational acceleration, and geomagnetic data. The inertial data collected from OxIOD was predominantly captured using an iPhone 7, while the ground truth was recorded through a Vicon motion capture system [33]. Among the datasets measured indoors, handheld data was used, which had the largest number of the data and recorded the largest change in orientation. To remove the error in the Vicon motion capture system, the initial 12 s and last 3 s of each sequence were discarded. Furthermore, in the OxIOD, an error of approximately 1%, which occurs when the normal value of the difference between the adjacent frames and orientation is 20° or more, was removed during training. During testing, the data containing errors were input for comparison with other models. The total number of training samples used was 54,885.

B. TRAINING

Tensorflow 1.13.1, Keras 2.3.1, and tfquaternion 0.1.6 were used in the training framework. Nvidia RTX 6000 was used as the graphics processing unit (GPU). An Adam optimizer was used with a learning rate of 0.0001. The batch size was 32, and the validation data was set to 10% of the training data. The network was trained for 500 epochs.

C. EVALUATIONS

As listed in Table 1, we divided the data into 17 training data and 6 testing data. For qualitative and quantitative evaluation of the trajectory, similar to other IONet, data from 0 to 20 s and 100 to 120 s were used to verify the performance at various instances. For quantitative evaluation of orientation, short-term data from 0 to 20 s and 100 to 120 s and long-term data from 0 to 100 s and 100 to 200 s were used to confirm the drift error in orientation. For the quantitative evaluation of position changes in the reference coordinate system, short-term data from 0 to 20 s and 100 to 120 s and long-term data from 0 to 100 s and 100 to 200 s were used to confirm the drift error for position change per unit time and the possibility of continuous localization. The evaluation was conducted on three models: 6-Axis IONet, 9-Axis IONet, and DO IONet.

TABLE 1. Training and testing splits.

TRAINING	TESTING
data1/seq1, seq3, seq4, seq7	data1/seq2, seq5, seq6
data2/seq1, seq2, seq3	data3/seq1
data3/seq2, seq3, seq4, seq5	data4/seq1, seq3
data4/seq2, seq4, seq5	data5/seq1
data5/seq2, seq3, seq4	

The parameter count of 6-Axis IONet is 1,161,735. The parameter count of 9-Axis IONet and DO IONet is 1,793,287, which requires more time for both training and evaluation.

V. RESULTS AND DISCUSSION

A. QUALITATIVE EVALUATION OF TRAJECTORY

In the trajectory evaluation, data from 0 to 20 s and data from 100 to 120 s of each sequence were used. Figs. 3 and 4 show the estimated trajectories by 6-axis IONet (a), 9-axis IONet (b), and DO IONet (c). In Fig. 3, the data between 0 and 20 s of the ''data1/seq2'' were used, and the data between 100 and 120 s of the ''data5/seq1'' were used in Fig. 4. In Fig. 3, it can be observed that the 6-axis IONet estimates the azimuth correctly; however, the error on the z-axis is extremely large. In contrast, the 9-axis IONet and DO IONet can precisely estimate the real trajectory. In Fig. 4, the trajectory of the 6-axis IONet is shifted to the top because of the drift in the pitch angle. The 9-axis IONet estimates the last point of the trajectory accurately; however, the shape is distorted. In contrast, it can be observed that DO IONet precisely estimates the shape of the real trajectory. Fig. 5 shows the norm of orientation error that occurs during the trajectory estimation of ''data1/seq2'' (a) and ''data5/seq1'' (b). In Fig. 5 (a), it can be observed that 6-Axis IONet diverges, while the error of 9-Axis IONet is generally less than 20 degrees. On the other hand, in (b), the 6-Axis IONet performs better. It can be confirmed that DO IONet generally estimates accurate poses and does not diverge even when errors occur.

B. QUANTITATIVE EVALUATION OF TRAJECTORY

To evaluate the proposed framework, we performed quantitative evaluation of each IONet using data between 0 and 20 s and 100 and 120 s of seven sets of testing data. Table 2 presents the root mean squared error (RMSE) of the trajectory estimated by each model. Of the 14 sequences, 6-axis IONet exhibited the lowest RMSE in 4, 9-axis IONet in 6, and DO IONet in 4 cases. The mean RMSE was lowest for the 9-axis IONet. The evaluation of the trajectory is a crucial indicator for evaluating IONet. However, as the process continuously calculates the difference between the estimated and real positions at each time, a detailed analysis is required to resolve the drift. Therefore, we conducted an additional evaluation of the orientation and position change in the reference coordinate system.

TABLE 2. Trajectory quantitative evaluation.

	6-AXIS [29] [m]		9-AXIS [30] [m]		DO IONET [m]	
	0s-20s	100s-120s	0s-20s	100s-120s	0s-20s	100s-120s
data1/seq2	0.681	0.625	0.168	**0.278**	**0.124**	1.039
data1/seq5	0.575	0.533	**0.219**	**0.255**	0.631	0.333
data1/seq6	0.615	0.568	0.143	0.335	**0.122**	**0.148**
data3/seq1	**0.353**	**0.397**	0.403	0.517	0.438	1.079
data4/seq1	**0.370**	**0.165**	0.413	0.292	4.032	0.472
data4/seq3	**0.518**	0.327	0.646	**0.163**	0.835	0.399
data5/seq1	**0.265**	0.413	0.292	0.697	0.818	**0.406**
mean	0.482	0.433	**0.326**	**0.362**	1.000	0.554
	0.458		**0.344**		0.777	

TABLE 3. Short-term quantitative evaluation of orientation.

	6-AXIS [29] [°]		9-AXIS [30] [°]		DO IONET [°]	
	0s-20s	100s-120s	0s-20s	100s-120s	0s-20s	100s-120s
data1/seq2	11.892	17.147	3.736	**7.519**	**3.725**	8.180
data1/seq5	10.105	18.316	**3.143**	12.705	4.594	**9.031**
data1/seq6	7.407	7.7254	**4.088**	7.651	4.314	**4.746**
data3/seq1	8.196	**9.089**	**4.586**	11.257	8.434	9.908
data4/seq1	**20.643**	6.942	21.203	8.178	30.725	**6.937**
data4/seq3	6.254	5.036	11.801	**3.864**	**5.383**	4.254
data5/seq1	**2.197**	**2.636**	3.518	14.623	4.498	3.413
mean	9.528	9.556	**7.439**	9.400	8.810	**6.639**
	9.542		8.4195		**7.7245**	

C. QUANTITATIVE EVALUATION OF THE ORIENTATION AND 3D TRANSLATION VECTOR IN THE REFERENCE COORDINATE SYSTEM

For a detailed evaluation of IONet, we performed a short-term quantitative evaluation using data from 0 to 20 s and 100 to 120 s and the long-term quantitative evaluation using data from 0 to 100 s and 100 to 200 s. Table 3 presents the RMSE of orientation for short-term evaluation. 6-axis IONet shows the lowest RMSE at 3, 9-axis IONet at 5, and DO IONet at 5. DO IONet has the highest performance in most sequences. The mean RMSE was the lowest in the proposed method.

Table 4, which lists the RMSE of the orientation for long-term evaluation, indicates that [29] and [30] are difficult to use as the estimated time increases. DO IONet exhibited the best performance in almost all sequences, and the mean RMSE was also the lowest. Furthermore, it did not diverge compared to the short-term orientation estimation.

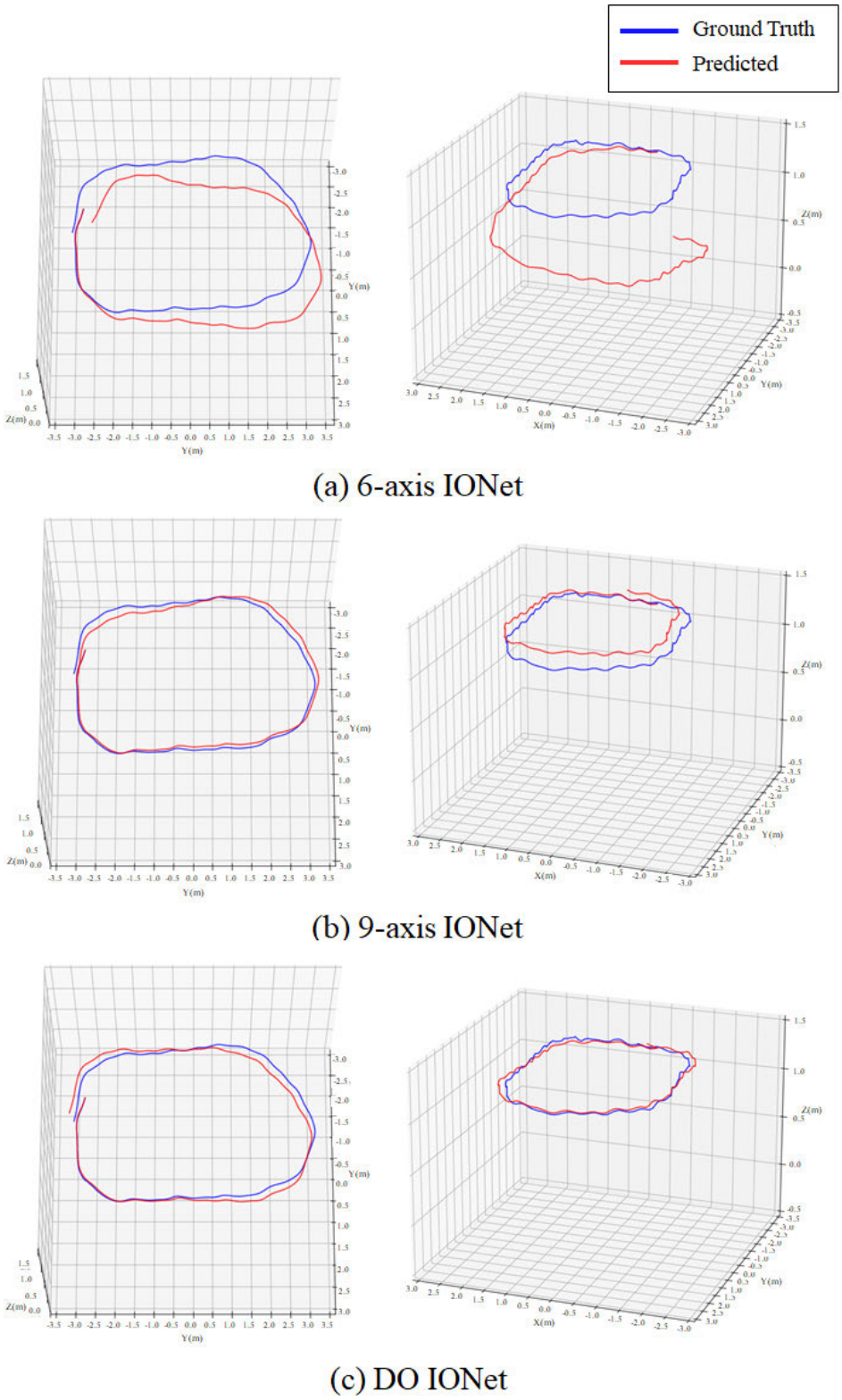

FIGURE 3. Top and side view of ground truth (blue) and predicted (red) trajectories of 'data1/seq2' dataset from 0sec to 20sec.

Tables 5 and 6 list the RMSE of the 3D translation vector in the reference coordinate system for short- and long-term, respectively. The 3D translation vector is the product of the rotation matrix and delta p, which is obtained using (5). It is an indicator that can confirm the drift by a vector that moves per unit time in the reference coordinate system.

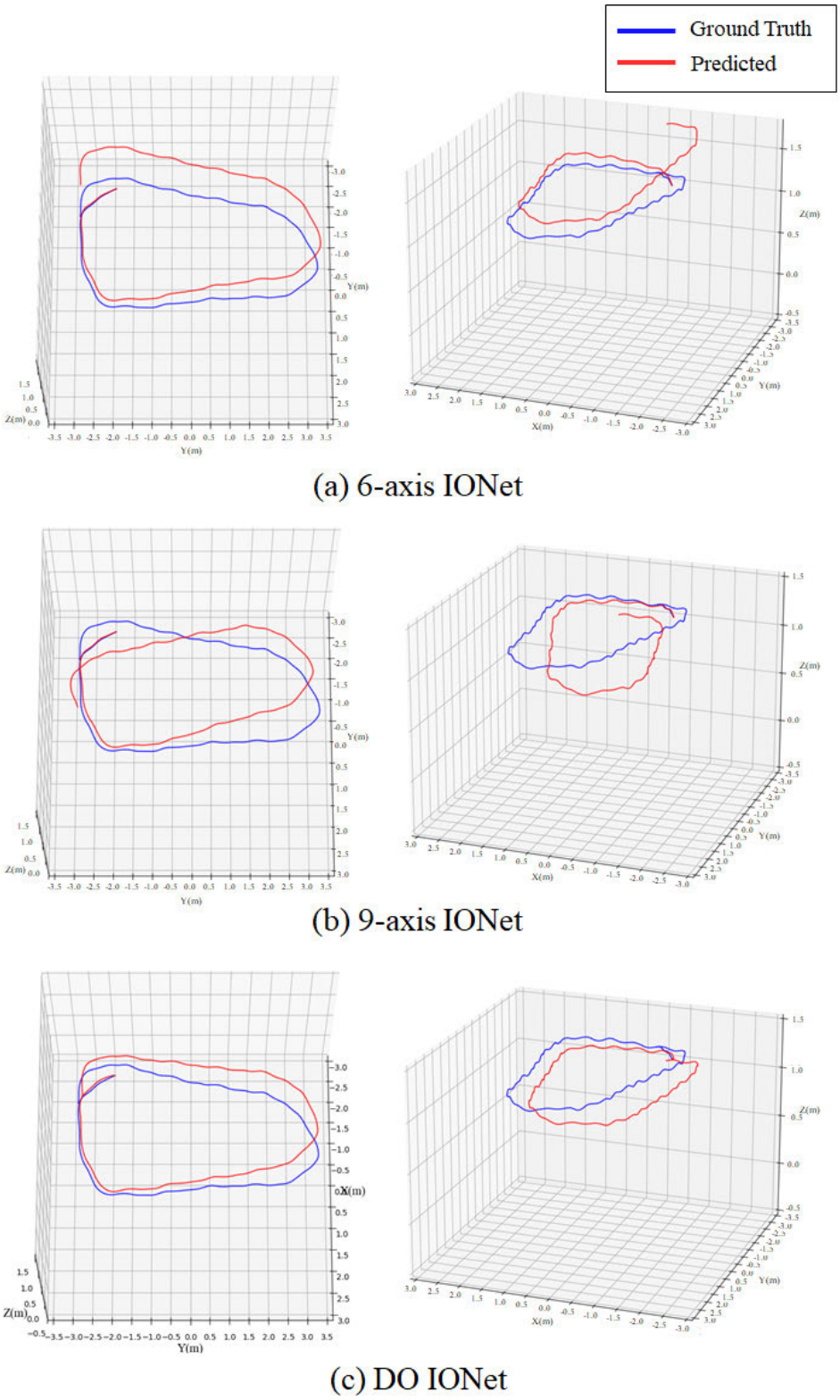

FIGURE 4. Top and side view of ground truth (blue) and predicted (red) trajectories of 'data5/seq1' dataset from 100sec to 120sec.

In Table 5, the 9-axis IONet is superior in nine sequences and mean RMSE among three models. However, DO IONet showed the best performance in most sequences, as indicated in Table 6, with a long estimation time. Moreover, DO IONet did not cause drift even if the estimation time was longer. The proposed framework does not excel in the trajectory

(a) 'data1/seq2' dataset from 0sec to 100sec

(b) 'data5/seq1' dataset from 100sec to 200sec

FIGURE 5. Norm of difference between ground truth and predicted orientation.

TABLE 4. Long-term quantitative evaluation of orientation.

	6-AXIS [29] [°]		9-AXIS [30] [°]		DO IONET [°]	
	0s-100s	100s-200s	0s-100s	100s-200s	0s-100s	100s-200s
data1/seq2	48.535	47.674	11.729	**8.738**	**3.912**	9.884
data1/seq5	55.092	43.974	14.842	18.593	**5.302**	**6.938**
data1/seq6	39.598	17.953	23.240	35.471	**6.457**	**5.938**
data3/seq1	21.087	21.591	28.374	34.219	**9.819**	**8.348**
data4/seq1	43.956	11.297	35.730	15.441	**25.004**	**6.913**
data4/seq3	16.451	6.549	27.106	27.617	**3.655**	**5.436**
data5/seq1	17.174	6.459	30.773	35.909	**4.962**	**4.821**
mean	34.556	22.214	24.542	25.141	**8.444**	**6.897**
	28.385		24.8415		**7.6705**	

TABLE 5. Short-term quantitative evaluation of 3D translation vector.

	6-AXIS [29] [m]		9-AXIS [30] [m]		DO IONET [m]	
	0s-20s	100s-120s	0s-20s	100s-120s	0s-20s	100s-120s
data1/seq2	0.0171	0.0167	**0.0063**	**0.0065**	0.0079	0.0142
data1/seq5	0.0222	0.0211	**0.0074**	**0.0105**	0.0127	0.0114
data1/seq6	0.0108	0.0144	**0.0065**	**0.0100**	0.0100	0.0104
data3/seq1	0.0116	**0.0136**	0.0134	0.0177	**0.0102**	0.0145
data4/seq1	**0.0096**	0.0064	0.0115	**0.0056**	0.0607	0.0129
data4/seq3	**0.0116**	0.0094	0.0175	**0.0055**	0.0148	0.0111
data5/seq1	0.0070	**0.0063**	**0.0070**	0.0226	0.0105	0.0088
mean	0.0128	0.0126	**0.0099**	**0.0112**	0.0181	0.0119
	0.0127		**0.0106**		0.0150	

estimation performance compared with the existing IONet, which estimates pose change; however, the drift error can be prevented in the pose change by appropriately using gravitational acceleration and geomagnetic sensor data.

TABLE 6. Long-term quantitative evaluation of 3D translation vector.

	6-AXIS [29] [m]		9-AXIS [30] [m]		DO IONET [m]	
	0s-100s	100s-200s	0s-100s	100s-200s	0s-100s	100s-200s
data1/seq2	0.0775	0.0762	0.0127	**0.0532**	**0.0095**	0.0546
data1/seq5	0.0818	0.0643	0.0146	0.0164	**0.0114**	**0.0115**
data1/seq6	0.0276	0.0246	0.0394	0.0577	**0.0118**	**0.0135**
data3/seq1	0.0322	0.0355	0.0450	0.0498	**0.0139**	**0.0130**
data4/seq1	**0.0202**	**0.0119**	0.0306	0.0153	0.0466	0.0126
data4/seq3	0.0146	**0.0095**	0.0294	0.0272	**0.0100**	0.0106
data5/seq1	0.0107	**0.0110**	0.0498	0.0543	**0.0100**	0.0121
mean	0.0378	0.0333	0.0316	0.0391	**0.0161**	**0.0183**
	0.0356		0.0354		**0.0172**	

VI. CONCLUSION

A novel 6-DOF initial odometry framework is proposed to address the need to set the reference orientation values when using IMU and the drift that diverges as the estimation time increases. The proposed approach estimates the position change and orientation of the sensor unit without constraints by inputting the linear acceleration, gyro data, gravitational acceleration, and geomagnetic data into the neural network in the reference coordinate system. The performance of the proposed method is evaluated in terms of the 3D trajectory, orientation, and position change using the OxIOD handheld sequence. The framework evaluation shows that compared with the existing IONet, the drift is minimized as the trajectory performance and estimation time are longer. Therefore, DO IONet can be applied to all types of odometry in which inertial odometry is used, such as visual inertial odometry, radar inertial odometry, and lidar inertial odometry, to set the initial orientation, which serves as the reference data for this method.

In the future, signal-based location recognition accuracy can be improved for both indoor and outdoor applications. We believe that the demand for odometry, which includes information on the user's trajectory, field of view, and position and orientation, will inevitably increase to improve the utilization of location information. However, most data used for inertial odometry are 2D and the existing 3D inertial data do not include 3D motion. Therefore, future research should focus on developing a stable and massive 3D inertial dataset.

REFERENCES

[1] S. A. S. Mohamed, M. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A survey on odometry for autonomous navigation systems," *IEEE Access*, vol. 7, pp. 97466–97486, 2019.

[2] E. J. Alqahtani, F. H. Alshamrani, H. F. Syed, and F. A. Alhaidari, "Survey on algorithms and techniques for indoor navigation systems," in *Proc. 21st Saudi Comput. Soc. Nat. Comput. Conf. (NCC)*, Apr. 2018, pp. 1–9.

[3] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, Apr. 2017.

[4] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2043–2050.

[5] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, "D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1278–1289.
[6] Y. S. Park, Y. Shin, and A. Kim, "PhaRaO: Direct radar odometry using phase correlation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2617–2623.
[7] D. Adolfsson, M. Magnusson, A. Alhashimi, A. J. Lilienthal, and H. Andreasson, "CFEAR radarodometry–conservative filtering for efficient and accurate radar odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 5462–5469.
[8] J. Zhang and S. Singh, "Low-drift and real-time LiDAR odometry and mapping," *Auton. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.
[9] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li, "LO-Net: Deep real-time LiDAR odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8465–8474.
[10] S. Y. Cho and C. G. Park, "A calibration technique for a two-axis magnetic compass in telematics devices," *ETRI J.*, vol. 27, no. 3, pp. 280–288, Jun. 2005.
[11] A. A. A. Rahni and I. Yahya, "Obtaining translation from a 6-DOF MEMS IMU—An overview," in *Proc. Asia–Pacific Conf. Appl. Electromagn.*, Dec. 2007, pp. 1–5.
[12] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on various inertial measurement unit (IMU) sensor applications," *Int. J. Signal Process. Syst.*, vol. 1, no. 2, pp. 256–262, 2013.
[13] S.-H. Lee, W.-Y. Kim, and D.-H. Seo, "Automatic self-reconstruction model for radio map in Wi-Fi fingerprinting," *Expert Syst. Appl.*, vol. 192, Apr. 2022, Art. no. 116455.
[14] J. H. Seong and D. H. Seo, "Selective unsupervised learning-based Wi-Fi fingerprint system using autoencoder and GAN," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1898–1909, Mar. 2020.
[15] J.-H. Seong and D.-H. Seo, "Environment adaptive localization method using Wi-Fi and Bluetooth low energy," *Wireless Pers. Commun.*, vol. 99, no. 2, pp. 765–778, Mar. 2018.
[16] P. G. Savage, "Strapdown inertial navigation integration algorithm design Part 1: Attitude algorithms," *J. Guid., Control, Dyn.*, vol. 21, no. 1, pp. 19–28, Jan. 1998.
[17] P. G. Savage, "Strapdown inertial navigation integration algorithm design Part 2: Velocity and position algorithms," *J. Guid., Control, Dyn.*, vol. 21, no. 2, pp. 208–221, Mar. 1998.
[18] Y. Wu, X. Hu, D. Hu, T. Li, and J. Lian, "Strapdown inertial navigation system algorithms based on dual quaternions," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 110–132, Jan. 2005.
[19] X. Chen, C. Shen, W.-B. Zhang, M. Tomizuka, Y. Xu, and K. Chiu, "Novel hybrid of strong tracking Kalman filter and wavelet neural network for GPS/INS during GPS outages," *Measurement*, vol. 46, no. 10, pp. 3847–3854, Dec. 2013.
[20] B. Cui, X. Chen, and X. Tang, "Improved cubature Kalman filter for GNSS/INS based on transformation of posterior sigma-points error," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2975–2987, Jun. 2017.
[21] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU," in *Proc. IEEE Int. Symp. Intell. Signal Process. (WISP)*, Aug. 2009, pp. 37–42.
[22] W. Kang and Y. Han, "SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization," *IEEE Sensors J.*, vol. 15, no. 5, pp. 2906–2916, May 2015.
[23] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proc. ACM Conf. Ubiquitous Comput.*, Sep. 2012, pp. 421–430.
[24] X. Chen, X. Zhang, M. Zhu, C. Lv, Y. Xu, and H. Guo, "A novel calibration method for tri-axial magnetometers based on an expanded error model and a two-step total least square algorithm," *Mobile Netw. Appl.*, vol. 27, no. 2, pp. 794–805, Apr. 2022.
[25] C. Chen, C. X. Lu, J. Wahlström, A. Markham, and N. Trigoni, "Deep neural network based inertial odometry using low-cost inertial measurement units," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1351–1364, Apr. 2021.
[26] H. Yan, Q. Shan, and Y. Furukawa, "RIDI: Robust IMU double integration," in *Proc. Eur. Comput. Vis. (ECCV)*, Sep. 2018, pp. 641–656.
[27] S. U. Kim, J. Lee, J. Yoon, S. Ko, and J. Kim, "Robust methods for estimating the orientation and position of IMU and MARG sensors," *Electron. Lett.*, vol. 57, no. 21, pp. 816–818, Oct. 2021.
[28] M. Zhang, M. Zhang, Y. Chen, and M. Li, "IMU data processing for inertial aided navigation: A recurrent neural network based approach," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 3992–3998.
[29] J. P. S. do Monte Lima, H. Uchiyama, and R.-I. Taniguchi, "End-to-end learning framework for IMU-based 6-DOF odometry," *Sensors*, vol. 19, no. 17, p. 3777, Aug. 2019.
[30] W.-Y. Kim, H.-I. Seo, and D.-H. Seo, "Nine-axis IMU-based extended inertial odometry neural network," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 115075.
[31] M. U. Hassan and J. Miura, "Neural network-based real-time odometry using IMU for crane system and its application to large-scale 3D mapping," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2022, pp. 1062–1068.
[32] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7482–7491.
[33] C. Chen, P. Zhao, C. Xiaoxuan Lu, W. Wang, A. Markham, and N. Trigoni, "OxIOD: The dataset for deep inertial odometry," 2018, *arXiv:1809.07491*.

HONG-IL SEO received the B.S. and M.S. degrees in electrical and electronics engineering from Korea Maritime and Ocean University, South Korea, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree in electrical and electronics engineering. His research interests include inertial odometry, computer vision, and signal processing.

JU-WON BAE received the B.S. degree in ocean engineering and the M.S. degree in electrical and electronics engineering from Korea Maritime and Ocean University, South Korea, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree in electrical and electronics engineering. His research interests include image captioning, computer vision, and anomaly detection.

WON-YEOL KIM received the B.S., M.S., and Ph.D. degrees in electrical and electronics engineering from Korea Maritime and Ocean University, South Korea, in 2014, 2016, and 2021, respectively. He was with the Artificial Intelligence Convergence Research Center for Regional Innovation, Korea Maritime and Ocean University. His research interests include positioning systems, sense networks, and embedded signal processing.

DONG-HOAN SEO received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyungpook National University, South Korea, in 1996, 1999, and 2003, respectively. Since 2004, he has been with Korea Maritime and Ocean University, where he is currently a Professor with the Division of Electronics and Electrical Information Engineering. His research interests include sense networks, signal processing, and computer vision.