The following publication Wang, Jingxian; Weng, Duojie; Qu, Xuanyu; Ding, Weihao; Chen, Wu(2023). A Novel Deep Odometry Network for Vehicle Positioning Based on Smartphone. IEEE Transactions on Instrumentation and Measurement, 72, 1-12 is available at https://doi.org/10.1109/TIM.2023.3240227.

A Novel Deep Odometry Network for Vehicle Positioning Based on Smartphone

Jingxian Wang, Duojie Weng, Xuanyu Qu, Weihao Ding, and Wu Chen

***Abstract*—Smartphone with multiple sensors integration has been widely used for navigation. The Inertial Measurement Unit (IMU) embedded in smartphones has been widely used for pedestrian navigation for counting steps. However, it is a challenge to measure the accurate velocity of the vehicle from the smartphone-embedded IMU data with a high noise level. Thus, current vehicle navigation with a smartphone relies substantially on the Global Navigation Satellite System (GNSS), which provides unreliable positions in urban dense areas due to the blockage and the reflection of GNSS signals. In this study, we propose a smartphone-based positioning method to improve vehicle positioning performance continuously in GNSS-degraded areas through the improvement of IMU velocity estimation. A convolutional neural network–Gated Recurrent Unit (CNN-GRU) combined deep learning odometry network, termed DeepOdo, is proposed to estimate the velocity of the vehicle with the IMU and barometer data as the input, rather than the traditional integral of the IMU measurements. Raw sensor data is utilized to boost the robustness. Labels of the DeepOdo are obtained from the integrated GNSS/IMU/Barometer solutions in the smartphone which significantly simplifies the dataset collection. In GNSS-denied areas, IMU, Barometer, and DeepOdo are integrated to provide accurate navigation solutions for the vehicle. Results of the proposed method show 73.14% and 98.33% improvements in horizontal and vertical directions respectively, compared with the Non-Holonomic Constraints (NHC) aided IMU. Finally, the DeepOdo network is deployed in Android smartphones to demonstrate that the proposed solution can work properly on the mobile platform.**

***Index Terms*—IMU, barometer, smartphone, deep learning odometry, vehicle positioning.**

I. INTRODUCTION

WITH the continuously growing demands of Intelligent Transportation Systems (ITS), the world has seen substantial development in vehicle positioning technologies in terms of accuracy and coverage [1]. Since the market penetration rate of vehicle built-in navigation systems was lower than 20% until 2020, smartphones are still the first choice for vehicle navigation. Global Navigation Satellite System (GNSS) module has been widely used in smartphones for positioning which provide global coverage with an accuracy of 5-10 m. Differential GNSS technologies can also be implemented in smartphones to improve the positioning accuracy to 2 m level [2]. Benefiting from the development of micro-electromechanical systems (MEMS) technology, the costs and power consumptions of chips reduce continuously. More and more sensors can be embedded into the smartphone, including GNSS [3], WiFi [4], Bluetooth Low Energy (BLE) [5], magnetometer [6], barometer [7], and Inertial Measurement Unit (IMU) [8]. Integrating sensors can further improve smartphone positioning accuracy.

IMU is a cost-controllable sensor that is not easily influenced by external conditions, which has been well documented in related research [9]. IMU was originally designed for ships, aircraft, and rockets which require very high accuracy while being insensitive to size and cost. The traditional inertial odometry method, such as the Inertial Navigation System (INS), transforms the values of the accelerometer and gyroscope from the body frame to the world navigation frame, which is then integrated to estimate the movement according to the mathematical model [10].

With the continuous advancement of MEMS technology, IMU becomes smaller, cheaper, and can be embedded into the smartphone. However, the cost and process of the smartphone built-in IMU dictate that it cannot meet the accuracy and stability requirements of traditional inertial navigation. Hence, in smartphone-based vehicle positioning, GNSS is the most widely used sensor in combination with IMU [11] to fill the gap when GNSS is not available. In GNSS-denied areas, many self-constrained techniques of IMU are proposed to mitigate the divergence of error. For instance, Heuristic Drift Reduction (HDR) is utilized to eliminate the drift of the gyroscope [12]. Zero velocity update (ZUPT) and zero integrated heading rate (ZIHR) are usually applied to control the divergence of the inertial navigation system (INS) when stationary [13]. Non-Holonomic Constraints (NHC) are used with an odometer to enhance the positioning precision while traveling [14].

High-precision odometry is a crucial and indispensable method to obtain 3D velocity assistance for vehicle positioning in dense urban areas. Different solutions have been developed with the use of a variety of sensors. Installing an odometer on the wheel is the simplest but most expensive way [15]. With the development of autonomous driving technology, RGB-D cameras have become the most commonly used sensors since they can provide high-quality depth images in real-time. According to their principles, they can be divided into two

This work was supported in part by the University Grants Committee of Hong Kong under the Research Impact Fund Grant R5009-21 and in part by the Research Institute of Land and Space, The Hong Kong Polytechnic University. *(Corresponding author: Duojie Weng).*

Jingxian Wang, Duojie Weng, Weihao Ding and Wu Chen are with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong (e-mail: jingxian.wang@connect.polyu.hk; duojieweng@gmail.com; weihao.ding@connect.polyu.hk; wu.chen@polyu.edu.hk).

Xuanyu Qu is with the Research Institute of Land and Space, The Hong Kong Polytechnic University, Hong Kong (e-mail: xuany.qu@connect.polyu.hk).

categories: one is using structured light, such as Occipital Structure Sensor and Microsoft Kinect V1; the other is using time-of-flight such as Microsoft Kinect V2 and Intel Realsense L515 [16]–[18]. The disadvantage is that all of them are relatively expensive. Binocular stereo vision is another solution to provide the traveled distance without requirements for depth cameras [19]; however, it does not work well in low-lighting conditions.

In addition to vision sensors, the data from other sensors are also integrated with IMU. For example, Walter et al. combine the gyroscope in the smartphone and the Controller Area Network (CAN) bus in the vehicle to obtain positioning results [20]. Gao et al. fuse the IMU and magnetometer and constrain their noises by the road features of the car park to realize positioning in underground parking [21]. Moussa et al. attach a smartphone to the steering wheel and use its accelerometer to estimate the steering angle which limits the IMU drifts during GNSS outages [22]. None of these methods are suitable for widespread dissemination.

Recently, researchers have studied the possibility to utilize deep learning techniques in inertial odometry. Sequential deep learning methods such as Long Short Term Memory (LSTM), which is an artificial recurrent neural network (RNN) architecture, are usually used to detect the characteristics of the time-series data. A two-layer LSTM network with 96 hidden nodes in each layer is presented as IONet [23] to estimate odometry by using raw IMU outputs of smartphones. The 2D displacement estimation can be provided in several different postures of the smartphone. However, the input IMU data cannot contain large biases and the postures of the smartphone are one-to-one matched with the trained networks. A new posture of the smartphone or the wrong selection of the trained network will lead to incorrect results. Chen et al. presented the MotionTransformer framework with Generative Adversarial Networks to improve the generalization ability of deep inertial odometry in new motion domains [24].

Tight Learned Inertial Odometry (TLIO) is introduced to estimate short-term 3D displacement and its uncertainty with raw IMU outputs by 1D version residual neural network (ResNet), which is an artificial convolutional neural network (CNN) [25]. These short-term data are fused tightly in a Kalman Filter to calibrate the pose, velocity, and sensor biases for PDR use. But the computation burden of TLIO is too high for the mobile platform. Wang et al. presented the Lightweight Learned Inertial Odometer (LLIO) to address the limits of computing on mobile platforms [26]. Compared to TLIO with ResNet, a residual multi-layer perceptron (ResMLP) is applied in the LLIO which has similar performance but significantly enhances efficiency.

Methods mentioned above try to solve the position directly. Compared with these methods, the Robust IMU Double Integration (RIDI) is unique in regressing the velocities from the history of IMU outputs with a trained deep neural network and correcting the linear accelerations. The corrected linear accelerations are integrated twice to estimate locations [27]. Similarly, Cortés et al. proposed a CNN-based deep-learning model to estimate the speed to constrain the classical INS [28].

As shown above, the majority of the deep learning based inertial odometry methods focus on solving the trajectory estimation problems for pedestrians. The architectures of these networks are designed according to the characteristics of pedestrian walking or running which is completely different from these of aerial vehicles or ground vehicles. To provide inertial odometry for the vehicle, Esfahani et al. proposed AbolDeepIO, a novel triple-channel LSTM network, to obtain characteristics from accelerometer, gyroscope, and time interval between two sampling times [29]. These characteristics are connected in the feature space to compute the changes in position and orientation. It is examined on a public dataset: the EuRoC micro aerial vehicle datasets [30] which shows its performance is better than IONet. The main issue is that AbolDeepIO is not integrated with the traditional INS to combine their strengths.

To integrate the INS with deep neural networks, Robust Inertial Navigation System on Wheels (RINS-W) was developed to detect the specific motion profiles of the vehicle, such as ZUPT, ZIHR, and NHC, to constrain INS by correcting system errors through Kalman filter [31], but the forward velocity errors are not constrained. To complement this weakness, Tang et al. presented the OdoNet, a CNN-based learning odometer model, to estimate the forward velocity [32]. Experiments show an improved result, but the input of OdoNet, which is the IMU data, needs to be compensated before use, and the datasets are required to be labeled by a highly accurate odometer. Existing methodologies of deep learning-based inertial odometry all have certain limitations. The robustness of the inertial odometry for the vehicle still needs to be studied.

More than half of the global population lives in cities, and there is an increasing demand for accurate vehicle positioning in urban areas where GNSS signals can be frequently blocked or reflected by buildings, flyovers, and tunnels [33]. Moreover, traffic jams happen frequently in these GNSS denied areas, which will cause long-period GNSS outages. Still, due to the restrictions of cost and size, the qualities of the smartphone's built-in sensors are relatively low. The low-quality IMU cannot last for sufficient time independently due to its unstable bias and large noise [34]. These factors pose great challenges to smartphone-based vehicle positioning systems.

Consequently, extra aiding constraints are necessary to enhance the performance of the smartphone-based vehicle positioning in GNSS-denied areas. For example, ZUPT and ZIHR constrain the divergence of INS results for stationary vehicles. NHC is settled to correct the lateral and vertical velocities for moving vehicles. The forward velocity constraint for moving vehicles is usually obtained by installing an odometer on the wheel or connecting to the CAN of the car. Nevertheless, it is inappropriate to require individual users to install additional hardware. Hence, the deep learning inertial odometry network based on IMU becomes a solution. Low-quality IMU usually contains large noises and time-variant biases. CNN is proficient in feature extraction of the spatial domain. It can be used to filter noise at the current moment.

Gated Recurrent Unit (GRU) can find out the relationships between CNN-extracted feature vectors in the time domain [35], which mitigates the effect of noise variations in the time domain on the forward velocity estimations. In addition, compared to LSTM, GRU has similar performance but cheaper computation costs. Hence, we propose a convolutional neural network-Gated Recurrent Unit (CNN-GRU) combined odometry network, called DeepOdo, to estimate the velocity of the vehicle based on the smartphone built-in sensors. The contributions of this paper are summarized as follows:

1) DeepOdo network estimates the velocity using temporal and spatial features, which are extracted from sensor data through the integration of CNN and GRU.
2) Raw data from barometer and IMU are taken into account by the proposed DeepOdo network, and they can significantly improve velocity estimation performance in terms of accuracy and robustness.
3) Labels of the training dataset are generated by the GNSS/IMU/Barometer integration method on the smartphone, which significantly simplifies the collection of the training dataset.
4) The integration of DeepOdo, IMU, and Barometer improves the positioning performance in GNSS-denied areas, and its usability has been demonstrated on normal Android smartphones.

The remaining parts of this paper are organized as follows: Section II details the architecture of the DeepOdo network and the framework of the integrated navigation system. Section III presents the experimental results for the precision and robustness evaluation of the proposed method. Section IV concludes the proposed algorithm.

II. PROPOSED METHOD

To increase the precision of smartphone-based vehicle navigation in the GNSS-denied areas, this paper proposes a novel deep odometry network, called DeepOdo, to assist vehicle positioning with the barometer. The whole procedure of the proposed algorithm is presented in Fig. 1. The GNSS module, barometer, and IMU embedded in the smartphone provide the required data to the system-on-a-chip (SoC) for calculation. At first, the INS is applied, and the raw data of IMU and barometer are inputs of DeepOdo to estimate the forward velocity of the vehicle as an odometer. Then, GNSS results and altitude based on the barometer are integrated with the INS in the extended Kalman filter (EKF). Depending on the vehicle running states, different constraints are chosen to assist the integration of GNSS/IMU. If the vehicle is stationary, ZUPT and ZIHR will be used as aids. When the vehicle is moving, the forward velocity provided by DeepOdo and the NHC will be added to the EKF. This section describes the architecture of DeepOdo at first. Then, the barometer-based altitude calculation method, the integration of GNSS/IMU/Barometer, and the assistance of NHC/DeepOdo/ZUPT/ZIHR are detailed in this section.

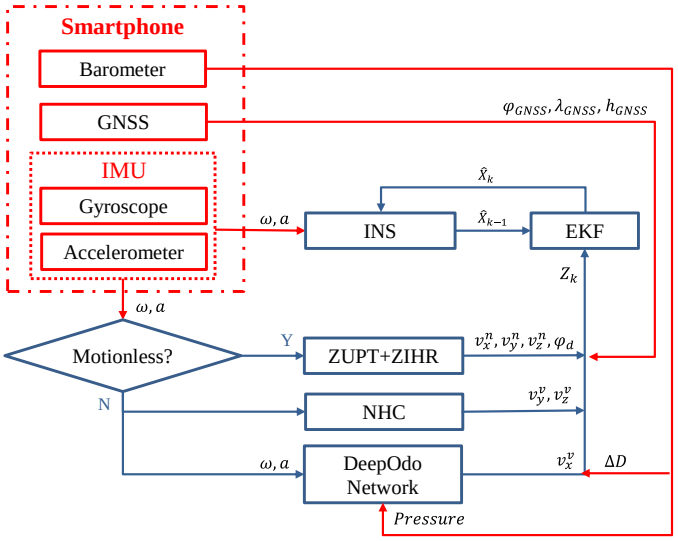


Fig. 1. Framework of DeepOdo network and barometer assisted vehicle positioning algorithm embedded into smartphone.

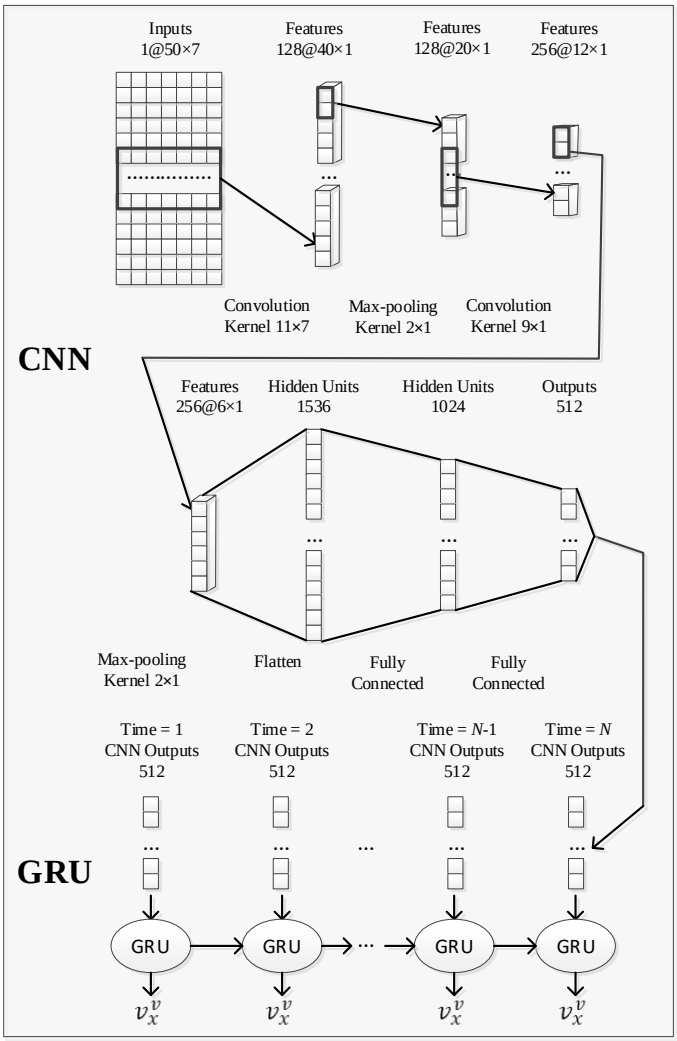


Fig. 2. Architecture of the proposed DeepOdo network. Convolution, Max-pooling, Flatten and Fully Connected are the hidden layers. Below the Convolution layer and Max-pooling are their kernel sizes ($height \times width$). The sizes of Inputs and Features are $channels@height \times width$. The sizes of Hidden Units and Outputs are $length$. These sizes are the optimal parameters obtained through experimentation.

A. DeepOdo Network

The architecture of the proposed DeepOdo network is shown in Fig. 2. In the CNN module, the inputs pass through 2 convolution layers and 2 max-pooling layers. After that, they are flattened and followed by two fully connected layers to generate the output (CNN extracted feature vectors) for the GRU module use. Considering the computational efficiency and vanishing gradient problems, Rectified Linear Unit (ReLU) is chosen as the activation function after the convolutions. Then, N CNN extracted feature vectors are transferred to the GRU module, which is a gating mechanism in RNN [35], to find out the relationships in the time domain for the forward velocity estimation. The kernel size of the convolution layer and max-pooling, and the units of the fully connected layer are demonstrated in Fig. 2. They are the optimal experimentally derived parameters. The CNN module and the GRU module in the proposed DeepOdo network are trained together. Hence, the inputs and outputs of the proposed DeepOdo network are shown as follows:

$$(w, a, P)_{N*50*7} \xrightarrow{F_\theta} (\hat{v}_{Odo}) \tag{1}$$

where N (50, 7) vectors consist of the input data of the DeepOdo network F_θ, N is the time window in GRU. Each vector contains 3-axis angular velocity w, 3-axis acceleration a, and 1-axis barometric pressure change P of 50 rows. In GNSS-denied areas, the noises of sensors are unable to be removed exactly. To get robust to noise, the raw data collected from the gyroscope, accelerometer, and barometer are used in the DeepOdo network. The output of DeepOdo is the estimated forward velocity of the vehicle $\hat{v}_{Odo}$.

According to the related works, IMU is the major input of the deep odometry networks [25], [32]. Time interval is used as an additional input in AbolDeepIO to reduce the influence of the existing sampling delay or absent problems in IMU data collection [29]. Since we have solved such problems in the data collection process, we did not use time intervals as an additional input. It is proved in [36] that the error of a CNN-Bidirectional LSTM integration method accumulates in the vertical direction. Since the change in barometric pressure is closely related to the change in altitude, the change in barometric pressure is considered as the additional input in our proposed DeepOdo network. The data frequency of the IMU and barometer used in the proposed integrated positioning system is 50 Hz. Since the update frequency of used EKF is 1 Hz, the output of DeepOdo is set at 1 Hz.

To obtain the output based on the inputs, finding the optimal parameter θ^* of DeepOdo network F is required. It can be calculated by minimizing the loss function on the training dataset:

$$\theta^* = \operatorname{argmin} loss\,(F_\theta(w, a, P), (\hat{v}_{Odo})) \tag{2}$$

The mean squared error (MSE) is used as the loss function during the network training. It is defined as:

$$\text{loss} = \frac{1}{N}\sum_{i=1}^{N} (v_{Odo} - \hat{v}_{Odo})^2 \tag{3}$$

where v_{Odo} is the truth of the forward velocity of the vehicle.

B. Barometer-based Absolute Altitude Calculation

Calculating the absolute altitude with a barometer usually requires the mean sea level (MSL) pressure at the current location and the pressure at the position. However, the MSL pressure varies with the weather and the location. The MSL of the current location is difficult to acquire. Furthermore, the smartphone barometer outputs are always influenced by systematic error. As a result, the absolute altitude is difficult to obtain directly [37]. To solve this problem, a methodology involving GNSS-aided absolute altitude calculation is proposed in this section. For altitudes within the troposphere, the formula that relates barometric pressure to relative altitude is:

$$\Delta h = \frac{T_0}{L}\left(\left(\frac{P-\Delta P}{P_0-\Delta P}\right)^{-\frac{R\cdot L}{g_0\cdot M}} - 1\right) \tag{4}$$

where Δh is the relative altitude between the current position and the previous position; T_0 is the reference temperature (288.15 K); L is temperature change over altitude (-0.0065 K/m); P is the pressure measured at the current position; P_0 is the pressure measured at the previous position; ΔP is the pressure systemic error of the barometer; R is the universal gas constant (8.3144598 $J/(mol \cdot K)$); M is the molar mass of Earth's air (0.0289644 kg/mol); g_0 is the gravitational acceleration (9.8099 m/s^2 in Hong Kong).

According to (4), ΔP can be calculated only if the ΔH is obtained. The altitude to calculate ΔH is provided by the GNSS/IMU integration from the smartphone in this paper. When the vehicle is running in an open area, the changes in altitudes and barometric pressures are recorded if the Vertical Dilution of Precision (VDOP) value of GNSS is less than the threshold which is set by experience. Until the number N_P of the recorded data is more than 3, ΔP can be estimated by minimizing the MSE shown in (5):

$$\Delta P^* = \operatorname{argmin} \frac{1}{N_P}\sum_{i=1}^{N_P} \left(\Delta h - \frac{T_0}{L}\left(\left(\frac{P-\Delta P}{P_0-\Delta P}\right)^{-\frac{R\cdot L}{g_0\cdot M}} - 1\right)\right)^2 \tag{5}$$

After getting the ΔP of the barometer, the position with the minimum VDOP value is selected as the previous position. Then, the absolute altitude can be calculated as follows:

$$h_{Baro} = h_{minVDOP} + \frac{T_0}{L}\left(\left(\frac{P-\Delta P}{P_{minVDOP}-\Delta P}\right)^{-\frac{R\cdot L}{g_0\cdot M}} - 1\right) \tag{6}$$

where h_{Baro} is the absolute altitude estimated by the barometer, $h_{minVDOP}$ is the altitude of the selected position and $P_{minVDOP}$ is the pressure of the selected position. The $h_{minVDOP}$ can achieve meter-level accuracy in an open area, and the relative altitude change of the smartphone can be accurately calculated by the barometric change with an error of less than one meter [38]. Therefore, the accuracy of h_{Baro} is meter-level.

C. GNSS/IMU/Barometer Integration

In this paper, the EKF is used to combine GNSS, IMU, and barometer with loose coupling. EKF is the most popular method for the integration of non-linear systems. Traditionally, the state equation of the extended Kalman filter is:

$$\dot{X}(t) = F(t)X(t) + G(t)W(t) \tag{7}$$

The state vector used in this system is shown below:

$$X = [\delta r \;\; \delta V \;\; \delta A \;\; \delta \omega \;\; \delta f]^T \tag{8}$$

where δr denotes the position error, δV denotes the velocity error, δA denotes the attitude error, $\delta\omega$ denotes the bias error of gyroscope and δf denotes the bias error of accelerometer. The North, East, Down (NED) coordinate is used in our system. Hence, the δr, δV, and δA can be specified as:

$$\delta r = [\delta\varphi \ \ \delta\lambda \ \ \delta h]^T \tag{9}$$

$$\delta V = [\delta V^n \ \ \delta V^e \ \ \delta V^d]^T \tag{10}$$

$$\delta A = [\delta A^n \ \ \delta A^e \ \ \delta A^d]^T \tag{11}$$

where $\delta\varphi, \delta\lambda, \delta h$ denotes the errors in latitude, longitude, and altitude.

The observation equation is:

$$Z(t) = H(t)X(t) + V(t) \tag{12}$$

Generally, the essential observation vector provided by GNSS is:

$$Z_{GNSS} = \begin{bmatrix} \varphi_{IMU} - \varphi_{GNSS} \\ \lambda_{IMU} - \lambda_{GNSS} \\ h_{IMU} - h_{GNSS} \end{bmatrix} \tag{13}$$

After the ΔP has been calculated in an open area, the observation vector based on the barometer is:

$$Z_{Baro} = [h_{IMU} - h_{Baro}] \tag{14}$$

where the accuracy of h_{Baro} is meter-level. Hence, it can constrain the integrated results in the same accuracy level through the EKF.

Since the smartphone's built-in sensors are very close to each other, the lever arms between these sensors are not considered in our algorithm, and the outputs of the built-in IMU are in the body frame specified by the smartphone.

D. Stationary Vehicle Constraints

When the vehicle stops at a red light or waits for someone on the side of the road, simple and efficient constraints such as ZUPT and ZIHR, are introduced to constrain the divergence of INS results. Theoretically, when the stationary state of the vehicle is detected, the velocities of the vehicle in three axes of the local-level frame should be all zeros. Simultaneously, the heading angle of the vehicle should be constant. Consequently, the additional observation vector is:

$$Z_{ZUPT} = \begin{bmatrix} \hat{v}_x^n - v_x^n \\ \hat{v}_y^n - v_y^n \\ \hat{v}_z^n - v_z^n \end{bmatrix} = \begin{bmatrix} \hat{v}_x^n - 0 \\ \hat{v}_y^n - 0 \\ \hat{v}_z^n - 0 \end{bmatrix} \tag{15}$$

$$Z_{ZIHR} = \left[\hat{A}^d - A^d_{stored}\right] \tag{16}$$

where A^d_{stored} is the heading stored when the vehicle becomes stationary.

The emphasis of ZUPT and ZIHR is that the motionless state of the vehicle needs to be detected. By analyzing experimental IMU data, results show that the vehicle can be considered static if the standard deviation of accelerometer and gyroscope values are lower than the empirical threshold. Every axis of the accelerometer and gyroscope can be expressed as:

$$STD = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i - \bar{\varepsilon}}, \ STD < T \tag{17}$$

where *STD* is the standard deviation, *T* is the threshold, *n* is the sample number in the moving window, ε_i is the current value, $\bar{\varepsilon}$ is the average value in the moving window.

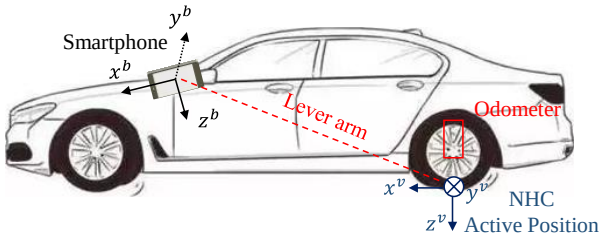


Fig. 3. Schematics of spatial alignment between the smartphone and the active position of the vehicle.

E. Moving Vehicle Constraints

As ZUPT and ZIHR are designed for stationary vehicles only, it is important to distinguish whether the vehicle is moving or not. The vehicle will not jump from the ground or slide laterally, and it only has the forward velocity when traveling normally. Therefore, the NHC can be settled. The lateral and vertical velocity should be zero, and the forward velocity can be estimated by the proposed DeepOdo. The velocity assistances of NHC with the odometer are designed in the vehicle frame. The actual velocity of the active position in the vehicle frame v^v_{AP} is shown as follows:

$$v^v_{AP} = [\hat{v}_{Odo} \ \ 0 \ \ 0]^T \tag{18}$$

As shown in Fig. 3, most vehicles steer with front wheels, and the rear wheels of the vehicle are considered as the installation position of the odometer and the active position (AP) of NHC to avoid the failure of NHC when the vehicle is steered [39]. Yet the smartphone is usually put in the front of the vehicle for user navigation with various mounting types. The spatial alignment between the smartphone built-in IMU in the body frame and the active position in the vehicle frame is required to be considered, which means the mounting angle and the lever arm between them. Therefore, when taking the spatial alignment between the IMU and the active position into consideration, the velocity in the active position can be calculated as:

$$\hat{\boldsymbol{v}}^v_{AP} = C^v_b \hat{C}^b_n \hat{\boldsymbol{v}}^n_{imu} + C^v_b\left(\omega^b_{nb} \times l^b_{AP}\right) \tag{19}$$

where C^v_b is the direction cosine matrix from the body frame of IMU to the vehicle frame based on the mounting angles, $\hat{C}^b_n$ is the estimated rotation matrix from the navigation frame to the body frame, $\hat{\boldsymbol{v}}^n_{imu}$ is the estimated velocity by IMU in the navigation frame, ω^b_{nb} is the angular rate vector of the body frame with respect to the navigation frame resolved in the body frame and l^b_{AP} is the lever arm vector from the active position to the IMU in the body frame. Therefore, the velocity error of the active position in vehicle frame $\boldsymbol{\delta v}^v_{AP}$ can be acquired by (19):

$$\delta v^v_{AP} = C^v_b C^b_n \delta v^n_{imu} - C^v_b C^b_n (v^n_{imu} \times)\delta A - C^v_b (l^b_{AP} \times)\delta\omega \tag{20}$$

Therefore, the observation vectors based on DeepOdo and NHC are:

$$Z_{Odo} = \left[\delta v^v_{AP,x}\right] = \left[\hat{v}^v_{AP,x} - v^v_x\right] \approx \left[\hat{v}^v_{AP,x} - \hat{v}_{Odo}\right] \tag{21}$$

$$Z_{NHC} = \begin{bmatrix} \delta v^v_{AP,y} \\ \delta v^v_{AP,z} \end{bmatrix} = \begin{bmatrix} \hat{v}^v_{AP,y} - v^v_y \\ \hat{v}^v_{AP,z} - v^v_z \end{bmatrix} \approx \begin{bmatrix} \hat{v}^v_{AP,y} - 0 \\ \hat{v}^v_{AP,z} - 0 \end{bmatrix} \tag{22}$$

III. EXPERIMENTS AND RESULTS

The procedure of the experiment and the equipment used in the experiment are detailed in this section. Then, the trained models by different datasets are compared. Furthermore, the enhancement of our proposed methodology in the GNSS-denied area is analyzed.

A. Dataset Preparation

To assess the performance of our proposed methodology, extensive experiments have been conducted to collect data in Hong Kong, a city with lots of high-rise buildings. As shown in Fig. 4, a navigation-grade GNSS/IMU system (iXblue-ATLANS-C) is installed on the car roof to provide centimeter-level ground truth of the velocities and positions. Huawei Mate 20 Pro is chosen as the experimental platform. The IMU embedded in this smartphone is InvenSense ICM20690 (Accelerometer noise: $100\,\mu g/\sqrt{Hz}$, Gyroscope noise: $\pm 4\,mdps/\sqrt{Hz}$), which is a low-cost MEMS IMU released in 2016. Most smartphone built-in IMUs have similar performance. The smartphone is installed under the windshield with a fixed mounting angle in Fig. 5. Since the mounting angle error is not the scope of this paper, the measured value of the mounting angle is set directly in our system for calculated C_b^v. The smartphone's built-in GNSS module records data at 1 Hz, and the IMU and barometer record data at 50 Hz.

The experimental routes are shown in Fig. 6. It contains 3 typical road environments, including congested streets, urban overhead roads, and suburban highways. There are total 25490 seconds of data collected for the dataset in 3 days. The first two days' data are used for network training (randomly selected 70% of the data as the training dataset, 30% as the validation dataset), and the last day's data is used for testing. The distributions of the velocities are demonstrated in Fig. 7. It shows that the velocity remains in the 15–25 m/s range for most of the time. It is very rare for velocity to be above 25 m/s.

The data of IMU and barometer is the input dataset for training DeepOdo. Traditionally, the true value of the forward velocities provided by ATLANS is used to label the dataset for training DeepOdo. To validate the ability of the smartphone for providing labels, the forward velocities calculated by the Mate 20 Pro built-in GNSS/IMU/Barometer integrated system are also used to label the dataset. Taking 1-hour data as an example in Fig. 8, the forward velocities calculated by the smartphone-based integration system have a similar trend to the true values but with a certain level of noise. The statistics show that the mean absolute error (MAE) of the velocity calculated by Mate 20 Pro data is 1.15 m/s.

B. Velocity Estimation Performance of DeepOdo

The proposed DeepOdo is implemented under the PyTorch framework and is trained by using an NVIDIA GTX 1080 GPU. The Adam, a first-order gradient-based optimizer [40] with a learning rate of 0.0002, is used in our training, and its learning rate decay is 0.8 in every 15 epochs. To prevent the influence of overfitting, 30% of dropouts are adopted in the training. These hyperparameters were obtained by using the grid search.

Fig. 4. Installation of navigation-grade GNSS/IMU system.

Fig. 5. Installation of smartphone on the vehicle.

Fig. 6. Experimental routes.

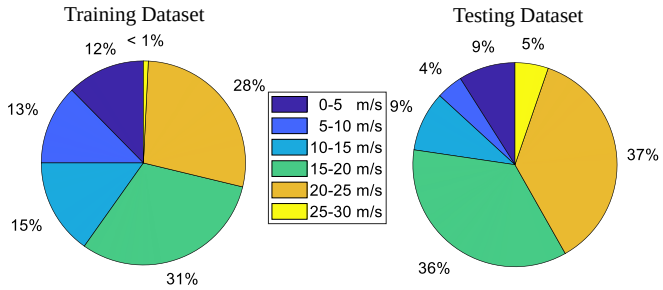


Fig. 7. Velocity distribution in training and testing datasets.

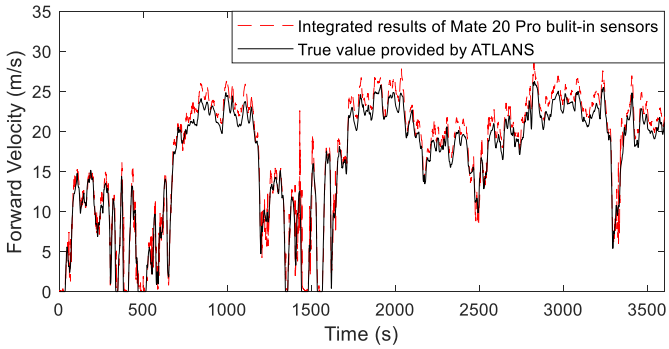
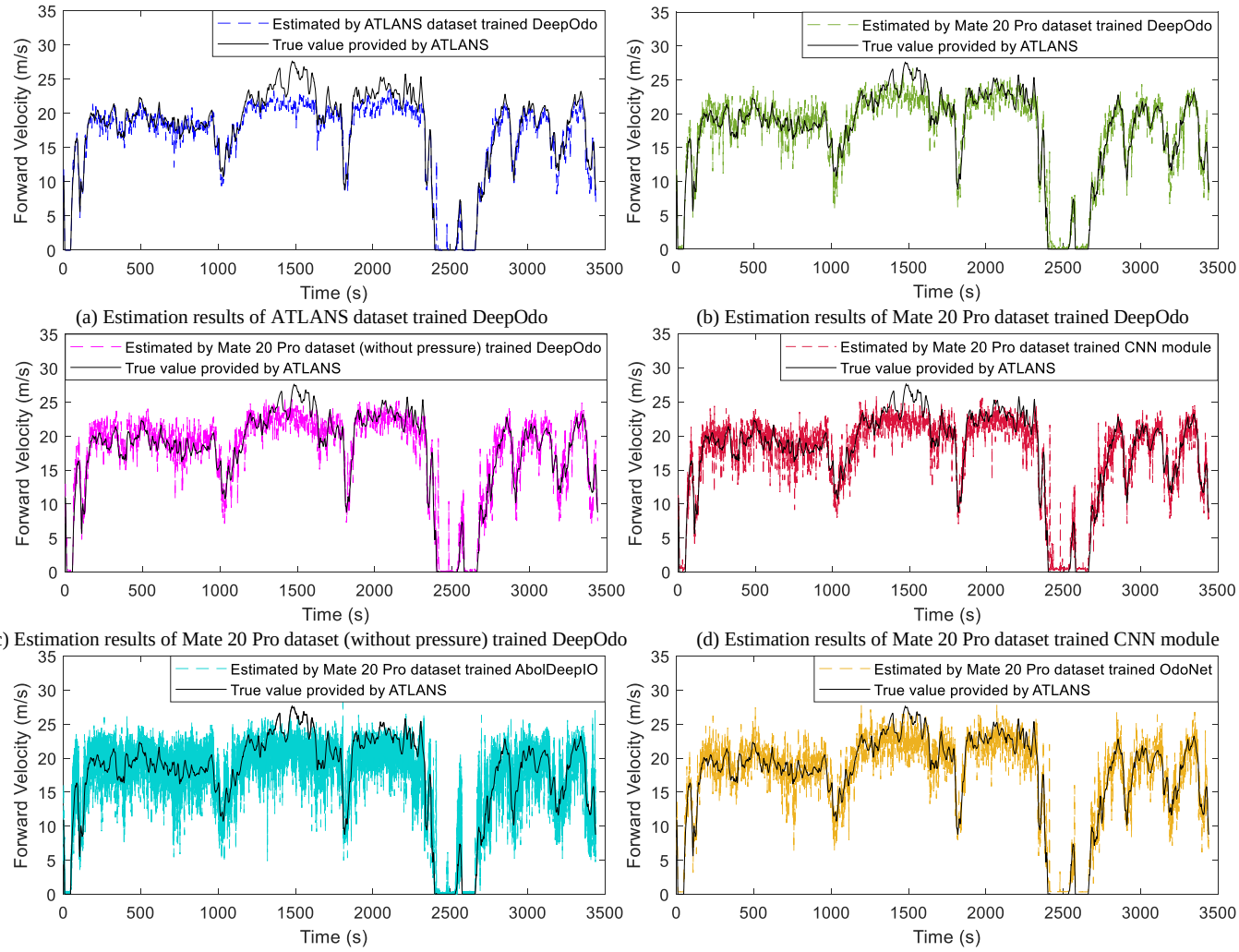


Fig. 8. Comparison of the datasets containing different labels.



(a) Estimation results of ATLANS dataset trained DeepOdo

(b) Estimation results of Mate 20 Pro dataset trained DeepOdo

(c) Estimation results of Mate 20 Pro dataset (without pressure) trained DeepOdo

(d) Estimation results of Mate 20 Pro dataset trained CNN module

(e) Estimation results of Mate 20 Pro dataset trained AbolDeepIO

(f) Estimation results of Mate 20 Pro dataset trained OdoNet

Fig. 9. The forward velocities estimated by different networks.

As mentioned before, two DeepOdo networks are trained with the ATLANS labeled dataset and the Mate 20 Pro labeled dataset separately to validate the ability of the smartphone for providing labels. Another DeepOdo network is trained with the Mate 20 Pro labeled dataset (without pressure) to determine if it is necessary to add the change in barometric pressure as the additional input in our proposed DeepOdo network. The CNN module in DeepOdo (input and output parts are modified) is also trained with the Mate 20 Pro labeled dataset to identify the importance of the GRU module in DeepOdo. To demonstrate the effectiveness of the proposed DeepOdo, we have compared DeepOdo network with two typical published deep odometer networks, AbolDeepIO (RNN model) [29] and OdoNet (CNN model) [32]. They are trained with same the Mate 20 Pro labeled dataset. For AbolDeepIO, there are 7 different architectures. We evaluated the performance of the AbolDeepIO07 architecture (window size 10) since it has the minimum overall MAE of the distance estimation. For OdoNet, we used the same parameters as given in the previous study [32].

As shown in Fig. 9, the forward velocities estimated by DeepOdo networks, CNN module, AbolDeepIO, and OdoNet are compared with the true value provided by ATLANS separately. These estimated velocities obtained from different deep neural networks follow the trend of the true value with different levels of variation, and they are all ineffective when the velocity of the vehicle is greater than 25 m/s. In Fig. 9 (a), the MAE of the evaluated velocities by the ATLANS dataset trained DeepOdo network is 1.31 m/s. In Fig. 9 (b), the MAE of the evaluated velocities by Mate 20 Pro dataset trained DeepOdo network is 1.34 m/s. It shows that the error of the labels provided by the Mate 20 Pro built-in GNSS/IMU/Barometer integrated system has a very small effect (2.3 % larger) on training the DeepOdo network. The smartphone can collect the training data and label them individually, which significantly reduces the complexity of training data collection. For the Mate 20 Pro dataset (without barometric pressure) trained DeepOdo network, the MAE of its evaluated velocities is 1.52 m/s in Fig. 9 (c), which demonstrates that including barometric pressure as an input improves MAE by about 12%. In Fig. 9 (d), the MAE of the evaluated velocities by the Mate 20 Pro dataset trained CNN module is 1.61 m/s, which is 20% larger than that of the proposed DeepOdo network. It proves that the GRU module has a significant effect on improving the accuracy of forward

velocity predictions. In Fig.9 (e) and (f), the MAE of the evaluated velocity by the Mate 20 Pro dataset trained AbolDeepIO and OdoNet are 2.25m/s and 1.81 m/s, which are 67% and 35% larger than that of the proposed DeepOdo network. It is demonstrated that the performance of our proposed CNN-GRU integrated network DeepOdo in velocity estimation is better than that of the RNN model AbolDeepIO and the CNN model OdoNet when using data from low-quality sensors.

The detailed comparison of the velocity errors in different velocity intervals is shown in Fig. 10, which is presented by the box plot. The short horizontal line inside the box represents the 50th percentile, and the top and bottom of the box represent the 75th and 25th percentiles, respectively. The top and bottom lines represent the 10th and 90th percentiles, respectively. The green triangles represent the mean values. It shows that the errors of the velocity estimated by different networks are mainly kept within 2m/s when the velocity of the vehicle is less than 25 m/s. On the contrary, when the velocities are above 25 m/s, these errors increase significantly. This is due to a lack of sufficient training data, as shown in Fig. 7, the data above 25 m/s is less than 1% in the training dataset. The mean values of the errors of the velocities evaluated by three DeepOdo networks are similar, but the variance of the errors of the velocities evaluated by the Mate 20 Pro dataset trained DeepOdo network is slightly larger than that of the ATLANS dataset trained DeepOdo network, and the variance of the errors is also slightly larger if the barometric pressures are not used as the input of the network or only use the CNN module of the proposed DeepOdo network. It also illustrates that the variance of the errors of the velocities evaluated by the Mate 20 Pro dataset trained DeepOdo network is smaller than that of the Mate 20 Pro dataset trained AbolDeepIO and OdoNet.

Currently, the training dataset and test dataset both contain data of the complete velocity range (0–30 m/s). To illustrate the generalization ability of the proposed DeepOdo network, we performed the cross-speed experiment by making the training dataset and the test dataset contain different velocity intervals. The ATLANS dataset is selected as the label since its velocities are more accurate. The data at the speed of 0-5, 10-15, 20-25 m/s are used as the training dataset and the data at the speed of 5-10, 15-20, 25-30 m/s are used as the test dataset. The velocity errors of the ATLANS dataset with the selected velocity intervals trained DeepOdo are compared with the conventional ATLANS dataset trained DeepOdo in Fig. 11. It shows that the mean values of the errors of the velocities evaluated by these two DeepOdo networks are similar, while the variance of the errors of the velocities evaluated by the categorized ATLANS dataset trained DeepOdo network is slightly larger than that of the conventional ATLANS dataset trained DeepOdo network. The MAE of the overall evaluated velocities by the categorized ATLANS dataset trained DeepOdo network is 1.46 m/s which is only 11% worse than the conventional ATLANS dataset trained DeepOdo network. Therefore, the generalization ability of our proposed DeepOdo network is acceptable.

We deployed the proposed DeepOdo network on three kinds

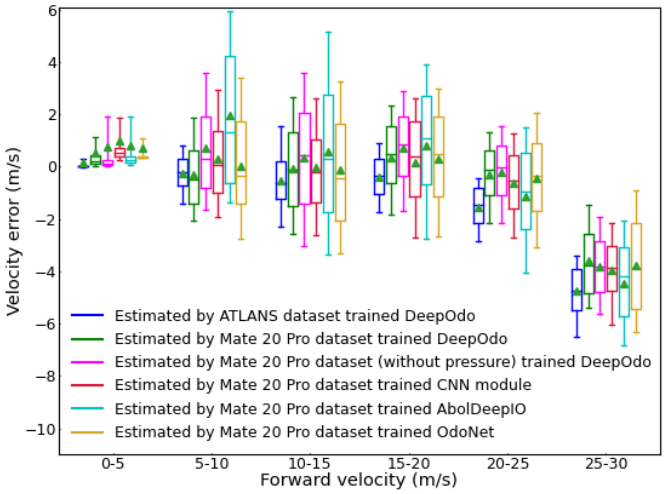


Fig. 10. Errors of the forward velocities estimated by different networks.

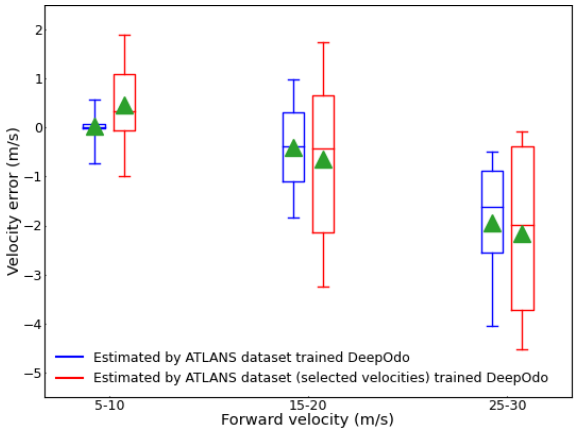


Fig. 11. Errors of the forward velocities estimated by different datasets trained DeepOdo networks.

TABLE I
Smartphones Specific Parameters and Their Experiment Results

Phone Type	Mate 20 Pro	Mate 40 Pro	Mi 12 Pro
Release Year	2018	2020	2021
Battery Capacity	4200 mAh	4300 mAh	4600 mAh
Average Elapsed Time	147 ms	139 ms	95 ms
Total Power Consumption	243 mAh	107 mAh	129 mAh

of smartphones to validate its performance. The specific parameters of these three phone types can be found in Table I. Mate 20 Pro was released in 2018, and its CPU (Kirin 980) performance is currently in the lower middle range. Mate 40 Pro was released in 2020, which has a newer process-manufactured CPU (Kirin 9000). Mi 12 Pro was released at the end of 2021, it is equipped with the latest process-manufactured CPU (Snapdragon 8 Gen 1). We carried out 10000 velocity estimations based on the DeepOdo network using the CPU of the smartphone in the test. Their estimated velocities are the same as the velocities estimated on the computer. According to Table I, the average elapsed times for each estimation of Mate 20 Pro, Mate 40 Pro, and Mi 12 Pro are 147 ms, 139 ms, and 95 ms separately. Since the EKF update frequency is 1 Hz, the forward velocity only needs to be estimated by the DeepOdo

network once a second. Hence, our proposed network can be used in real-time in the EKF if the elapsed time is less than 1000 ms. The total power consumptions for the 10000 calculations of Mate 20 Pro, Mate 40 Pro, and Mi 12 Pro are 243 mAh, 107 mAh, and 129 mAh respectively, which are 5.78%, 2.49%, and 2.80% of their battery capacities. Therefore, it is expected our algorithm can work properly on most of the new off-the-shelf smartphones today and that its power consumption is acceptable.

C. IMU/Barometer/DeepOdo Integration Results in GNSS-denied Area

To evaluate the enhancement of the integration of DeepOdo and barometer during a GNSS outage, the Central–Wan Chai Bypass Tunnel in Hong Kong is chosen as the test area. As shown in Fig. 12, this GNSS-denied tunnel is about 4 kilometers and it does not overlap with the trajectory of the training data. The time spent in the experiment driving through the tunnel is 207s. Since the GNSS is blocked, four configurations of different vehicle positioning systems are chosen to compare their performances. They are IMU, IMU/DeepOdo, IMU/Barometer/DeepOdo, and IMU/Barometer/Odometry, where the IMU stands for the INS with the aid of NHC and ZUPT/ZIHR, and odometry values are provided by ATLANS. The evaluated forward velocity by the Mate 20 Pro dataset trained DeepOdo is demonstrated in Fig. 13. A Savitzky–Golay filter is used to smooth the estimated velocities [41]. Its MAE is 0.87 m/s after smoothing.

The horizontal performances of these configurations over time can be seen in Fig. 14. It shows that the results of IMU with self-constraints gradually diverge with time. The integration of IMU and forward velocity provided by the proposed DeepOdo network or the odometry can solve this problem. Compared with the odometry provided by ATLANS, there are some errors in the estimated forward velocity by DeepOdo, but the integrated results of two different configurations (IMU/Barometer/DeepOdo and IMU/Barometer/Odometry) have a similar trajectory in the first 120 s. In the horizontal direction, the addition of the barometer has little effect on the results. However, as shown in Fig. 15, the altitude results of IMU/Barometer/DeepOdo (green line) and IMU/Barometer/Odometry (purple line) are basically the same as the ground truth. It demonstrates that the barometer can guarantee the altitude accuracy throughout the entire time whereas the DeepOdo can only smooth the altitude dispersion.

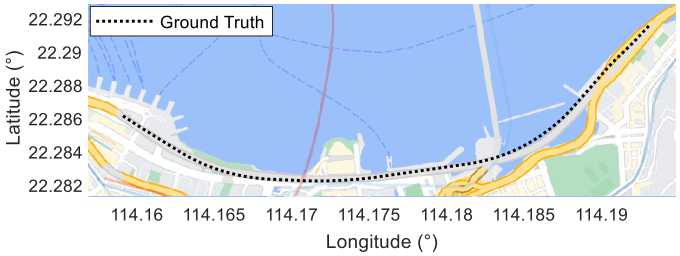


Fig. 12. Schematic of the Central–Wan Chai Bypass Tunnel.

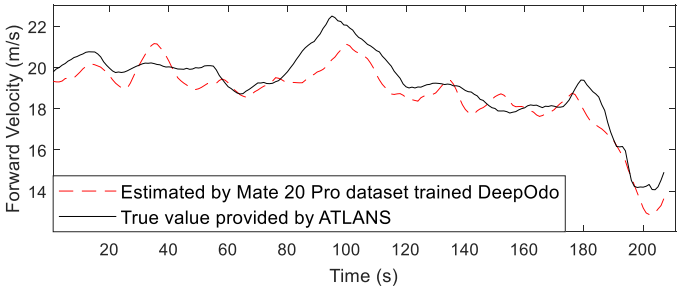


Fig. 13. Estimation of the forward velocity by the DeepOdo network.

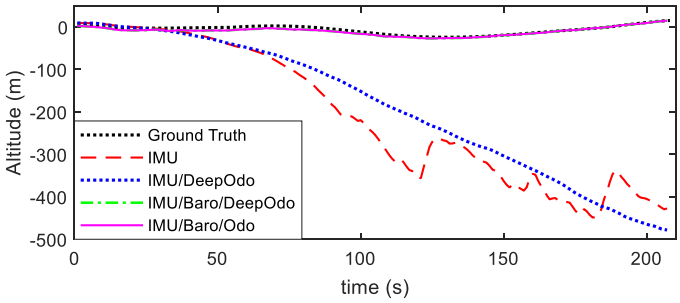


Fig. 15. The positioning performance of the tunnel test in vertical direction. The results of IMU/Baro/DeepOdo and IMU/Baro/Odo are basically the same (the green line is mostly covered by the purple line).

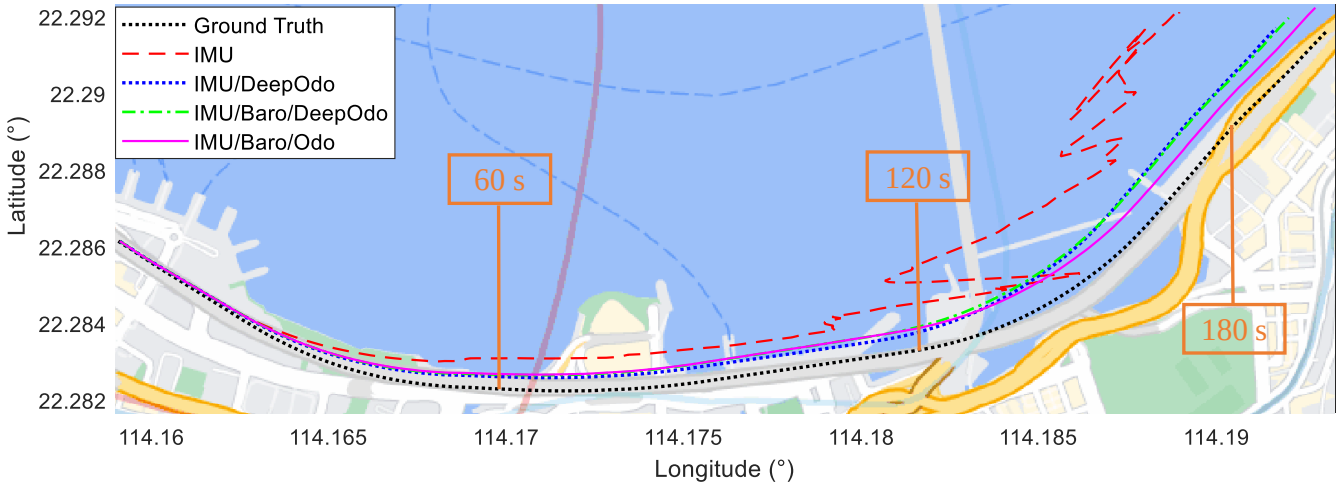


Fig. 14. The positioning performance of the tunnel test in horizontal direction. In the first 120 s, the results of IMU/Baro/DeepOdo and IMU/Baro/Odo are basically the same (the green line is mostly covered by the purple line). After 120 s, the results of IMU/Baro/DeepOdo and IMU/DeepOdo are close (the green line is adjacent to the blue line).

The reason is the altitude error calculated by the barometer data mainly relevant to the error of initial altitude and the accuracy of barometer data. It does not accumulate with time like the error of DeepOdo.

The detailed MAE of different kinds of system configurations is illustrated in Table II. It shows that the MAE of IMU results are 241.6 m and 204.6 m in horizontal and vertical separately, which is quite large. The aid from the forward velocity calculated by DeepOdo can be combined with NHC to constrain the error only in the horizontal direction effectively. The MAE in the horizontal direction can be reduced to 68.9 m (71.5 %); however, it only decreases by 10.9 % in the vertical direction, an insignificant improvement. With the help of the barometer, the vertical error can be well controlled under 5 m in the last 2 configurations, whereas insufficiently accurate mounting angles lead to a small increase in horizontal errors in the first 120 s. Compared to the MAE of IMU/Barometer/Odometry, the MAE of IMU/Barometer/DeepOdo is 0.9 m better in the first minute, and 1.7 m worse in the first 120 s. Eventually, the proposed algorithm which integrates IMU/Barometer/DeepOdo shows improvements of 73.14% and 98.33% in horizontal and vertical directions respectively, whereas the enhancements of IMU/Barometer/Odometry integration are 78.06% (horizontal) and 98.33% (vertical). It proves that our proposed DeepOdo network can be an alternative to the traditional wheeled odometer within a certain period.

TABLE II
The MAE of Different System Configurations in Tunnel

Time (s)	IMU		IMU/ DeepOdo		IMU/Baro/ DeepOdo		IMU/Baro/ Odo	
	H (m)	V (m)	H (m)	V (m)	H (m)	V (m)	H (m)	V (m)
60	67.2	13.8	20.0	14.4	21.9	5.1	22.8	5.1
120	162.6	96.1	37.3	66.2	39.9	5.1	38.2	5.1
180	211.1	174.7	59.6	141.6	58.5	3.8	49.6	3.8
207	241.6	204.6	68.9	182.3	64.9	3.4	53.0	3.4

Baro = Barometer, Odo = Odometry, H = Horizontal, V = Vertical.

IV. Conclusion

This paper proposed a CNN-GRU combined deep learning odometry network (DeepOdo) to estimate the forward velocity and assist the vehicle positioning with the embedded barometer. The proposed DeepOdo network learns to extract characteristics from the IMU and barometer to predict the velocities of vehicles. Comprehensive experiments have been conducted in Hong Kong to demonstrate the effectiveness and robustness of DeepOdo and the proposed integrated system in GNSS-denied areas:

1) The MAE of the evaluated velocities by the Mate 20 Pro dataset trained CNN module is 1.61 m/s, which is 20% larger than that of the proposed DeepOdo network. The MAE of the evaluated velocity by the Mate 20 Pro dataset trained AbolDeepIO and OdoNet are 2.25m/s and 1.81 m/s, which are 67% and 35% larger than that of the proposed DeepOdo network. It is demonstrated that the combination of CNN and GRU can better filter the noises of low-quality sensors for velocity estimation than the individual RNN model (AbolDeepIO) and the CNN model (CNN module of DeepOdo or OdoNet).
2) Compared with the Mate 20 Pro dataset (without barometric pressure) trained DeepOdo network (1.52 m/s), including barometric pressure as an input improves MAE by about 12%. The MAE of the overall evaluated velocities by the categorized ATLANS dataset trained DeepOdo network is only 11% worse than the conventional ATLANS dataset trained DeepOdo network. It proves that the generalization ability of our proposed DeepOdo network is acceptable. The MAE of the evaluated velocities by the Mate 20 Pro dataset trained DeepOdo network is 2.3% larger than that of the ATLANS dataset trained DeepOdo network, which suggests that the smartphone can collect the training data and label them individually.
3) In GNSS-denied areas, results show that the IMU positioning errors are reduced by 71.5% and 10.9% in horizontal and vertical directions respectively with the aid of the forward velocity estimated by DeepOdo, compared with the positioning results provided by IMU only. By using the barometer data, the errors can be further reduced by about 5.8% and 98.1% in horizontal and vertical directions respectively.
4) The DeepOdo network is deployed in three kinds of smartphones that are released from 2018 to 2021. Their average elapsed times of each epoch are less than 150 ms. The required elapsed time should be less than 1000 ms. It demonstrates the proposed algorithm can work properly on mobile platforms.

In summary, our proposed DeepOdo network can estimate the forward velocity of vehicles to enhance the performance of the integrated positioning system with the potential to replace the traditional wheeled odometer. This provides the basis for smartphone-based lane-level navigation in complex environments. The current shortcoming of our approach is the mounting angle of the smartphone in collecting the data is relatively homogeneous. In the future, smartphones will collect data at different mounting angles for model training.

References

[1] J. Liu and G. Guo, "Vehicle Localization During GPS Outages With Extended Kalman Filter and Deep Learning," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1-10, 2021, Art no. 7503410, doi: 10.1109/TIM.2021.3097401.

[2] D. Weng, X. Gan, W. Chen, S. Ji, and Y. Lu, "A New DGNSS Positioning Infrastructure for Android Smartphones," *Sensors*, vol. 20, no. 2, p. 487, Jan. 2020, doi: ARTN 48710.3390/s20020487.

[3] R. Xu, W. Chen, Y. Xu, S. Ji, and J. Liu, "Improved GNSS-based indoor positioning algorithm for mobile devices," *GPS Solut.*, vol. 21, no. 4, pp. 1721–1733, Jul. 2017, doi: 10.1007/s10291-017-0647-0

[4] Q. Zeng, J. Wang, Q. Meng, X. Zhang, and S. Zeng, "Seamless pedestrian navigation methodology optimized for indoor/outdoor detection," *IEEE Sensors J.*, vol. 18, no. 1, pp. 363-374, Oct. 2017.

[5] H. Luo et al., "Integration of GNSS and BLE Technology With Inertial Sensors for Real-Time Positioning in Urban Environments," *IEEE Access*, vol. 9, pp. 15744-15763, Jan. 2021, doi: 10.1109/ACCESS.2021.3052733.

[6] Y. Dong, T. Arslan and Y. Yang, "Magnetic Disturbance Detection for Smartphone-Based Indoor Positioning Systems With Unsupervised Learning," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1-11, 2022, Art no. 2506411, doi: 10.1109/TIM.2022.3163145.
[7] M. Won, A. Mishra, and S. H. Son, "HybridBaro: Mining Driving Routes Using Barometer Sensor of Smartphone," *IEEE Sensors J.*, vol. 17, no. 19, pp. 6397-6408, Oct 2017, doi: 10.1109/jsen.2017.2734919..
[8] J. Wahlstrom, I. Skog, P. Handel, and A. Nehorai, "IMU-Based Smartphone-to-Vehicle Positioning," *IEEE Trans. Intell. Veh.*, vol. 1, no. 2, pp. 139-147, Jun. 2016, doi: 10.1109/tiv.2016.2588978.
[9] Y. Pang et al., "Low-Cost IMU Error Intercorrection Method for Verticality Measurement," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1-14, 2021, Art no. 2515814, doi: 10.1109/TIM.2021.3120447.
[10] A. H. Mohamed and K. P. Schwarz, "Adaptive Kalman filtering for INS GPS," *J Geodesy*, vol. 73, no. 4, pp. 193-203, May 1999, doi: 10.1007/s001900050236.
[11] X. J. Niu, Q. Zhang, Y. Li, Y. H. Cheng, and C. Shi, "Using Inertial Sensors of iPhone 4 for Car Navigation," in *Proc. IEEE/ION Position Location Navigat. Symp. (PLANS)*, Myrtle Beach, SC, USA, Apr. 2012, pp. 555–561.
[12] J. Borenstein and L. Ojeda, "Heuristic reduction of gyro drift in vehicle tracking applications," *Int. J. Veh. Inf. Commun. Syst.*, vol. 2, no. 1/2, 2009, doi: 10.1504/ijvics.2009.027747.
[13] C. Y. Liu, C. A. Lin, K. W. Chiang, S. C. Huang, C. C. Chang, and J. M. Cai, "Performance Evaluation of Real-time MEMS INS/GPS Integration with ZUPT/ZIHR/NHC for Land Navigation," in *Proc.12th Int. Conf. ITS Telecom.*, Taipei, Taiwan, Nov. 2012, pp. 584-588.
[14] X. Niu, S. Nassar, and N. EL‐SHEIMY, "An accurate land‐vehicle MEMS IMU/GPS navigation system using 3D auxiliary velocity updates," *Navig.*, vol. 54, no. 3, pp. 177-188, Aug. 2007.
[15] K. W. Chiang et al., "Assessment for INS/GNSS/Odometer/Barometer Integration in Loosely-Coupled and Tightly-Coupled Scheme in a GNSS-Degraded Environment," *IEEE Sensors J.*, vol. 20, no. 6, pp. 3057-3069, Mar. 2020, doi: 10.1109/Jsen.2019.2954532.
[16] M. G. Diaz, F. Tombari, P. Rodriguez-Gonzalvez, and D. Gonzalez-Aguilera, "Analysis and evaluation between the first and the second generation of RGB-D sensors," *IEEE Sensors J.*, vol. 15, no. 11, pp. 6507-6516, Jul. 2015.
[17] J. A. Albert, V. Owolabi, A. Gebel, C. M. Brahms, U. Granacher, and B. Arnrich, "Evaluation of the pose tracking performance of the azure kinect and kinect v2 for gait analysis in comparison with a gold standard: A pilot study," *Sensors*, vol. 20, no. 18, p. 5104, Sep. 2020.
[18] F. Lourenço and H. Araujo, "Intel RealSense SR305, D415 and L515: Experimental Evaluation and Comparison of Depth Estimation," in *Proc. 16th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, ELECTR NETWORK, Feb. 2021, pp. 362-369.
[19] S. Chen, C.-Y. Wen, Y. Zou, and W. Chen, "Stereo visual inertial pose estimation based on feedforward-feedback loops," 2020. [Online]. Available: arXiv:2007.02250.
[20] O. Walter, J. Schmalenstroeer, A. Engler, and R. Haeb-Umbach, "Smartphone-Based Sensor Fusion for Improved Vehicular Navigation," in *Proc. 10th IEEE Workshop Positioning Navigat. Commun.*, Dresden, Germany, Mar. 2013, pp. 1–6.
[21] R. P. Gao, G. J. Luo, and F. Ye, "VeMap: Indoor Road Map Construction via Smartphone-based Vehicle Tracking," in *Proc. 59th IEEE GLOBECOM*, Washington, DC, USA, Dec. 2016, pp. 1-6.
[22] M. Moussa, A. Moussa, and N. El-Sheimy, "Steering Angle Assisted Vehicular Navigation Using Portable Devices in GNSS-Denied Environments," *Sensors*, vol. 19, no. 7, Apr. 2019, doi: ARTN 161810.3390/s19071618.
[23] C. Chen, X. Lu, A. Markham, and N. Trigoni, "IONet: Learning to cure the curse of drift in inertial odometry," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2018, pp. 6468-6476.
[24] C. Chen et al., "MotionTransformer: Transferring Neural Inertial Tracking between Domains", in *Proc. 32nd AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, Feb. 2019, pp. 8009-8016.
[25] W. Liu et al., "TLIO: Tight Learned Inertial Odometry," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5653-5660, Oct. 2020.
[26] Y. Wang, J. Kuang, and X. Niu, "LLIO: Lightweight Learned Inertial Odometer," 2021. [Online]. Available: https://doi.org/10.36227/techrxiv.16408383.v1
[27] H. Yan, Q. Shan, and Y. Furukawa, "RIDI: Robust IMU Double Integration," in *Proc. Eur. Conf. Comput. Vision*, Munich, Germany, Sep. 2018, pp. 621–636.
[28] S. Cortés, A. Solin and J. Kannala, "Deep learning based speed estimation for constraining strapdown inertial navigation on smartphones," in *Proc. IEEE 28th Int. Workshop Mach. Learn. Signal Process.*, Aalborg, Denmark, Sep. 2018, pp. 1–6.
[29] M. A. Esfahani, H. Wang, K. Wu, and S. Yuan, "AbolDeepIO: A novel deep inertial odometry network for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst*, vol. 21, no. 5, pp. 1941-1950, May 2020.
[30] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157-1163, Sep. 2016, doi: 10.1177/0278364915620033.
[31] M. Brossard, A. Barrau, and S. Bonnabel, "RINS-W: Robust Inertial Nnavigation System on Wheels," in *Proc. Int. Conf. Intell. Robots Syst.*, Macau, China, Nov. 2019, pp. 2068–2075.
[32] H. Tang, X. Niu, T. Zhang, Y. Li, and J. Liu, "OdoNet: Untethered Speed Aiding for Vehicle Navigation Without Hardware Wheeled Odometer," 2021. [Online]. Available: arXiv:2109.03091.
[33] H. Luo, D. J. Weng, W. Chen, "An Improved Shadow Matching Method for Smartphone Positioning," *Geomatics Inf. Sci. Wuhan Univ.*, vol. 46, no. 12, pp. 1907-1915, Dec 2021, doi: 10.13203/j.whugis20210275.
[34] N. El-Sheimy, H. Hou, and X. Niu, "Analysis and Modeling of Inertial Sensors Using Allan Variance," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 1, pp. 140-149, Dec. 2007, doi: 10.1109/tim.2007.908635.
[35] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014. [Online]. Available: arXiv:1409.1259.
[36] J. P. S. D. Lima, H. Uchiyama, and R. Taniguchi, "End-to-End Learning Framework for IMU-Based 6-DOF Odometry," *Sensors*, vol. 19, no. 17, Sep. 2019, doi: ARTN 377710.3390/s19173777.
[37] H. B. Ye, L. Sheng, T. Gu, and Z. Q. Huang, "SELoc: Collect Your Location Data Using Only a Barometer Sensor," *IEEE Access*, vol. 7, pp. 88705-88717, Jun. 2019, doi: 10.1109/Access.2019.2925460.
[38] H. Ye, K. Dong, and T. Gu, "HiMeter: Telling You the Height Rather than the Altitude", *Sensors*, vol. 18, no. 6, pp. 1712, May 2018.
[39] Q. Zhang, Y. Q. Hu, and X. J. Niu, "Required Lever Arm Accuracy of Non-Holonomic Constraint for Land Vehicle Navigation," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8305-8316, Aug. 2020, doi: 10.1109/Tvt.2020.2995076.
[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: arXiv:1412.6980.
[41] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627-1639, Jul. 1964.

Jingxian Wang received his B.S. degrees and M.S. degrees from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015 and 2018. He is currently pursuing the Ph.D. degree with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong.

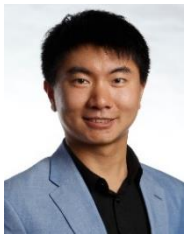
His current research interests include multi-sensors fusion, indoor position, pedestrian and vehicle navigation.

Duojie Weng received the B.S. and M.S. degrees in electrical engineering from Hohai University, Nanjing, China in 2007 and 2010, and the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, in 2016.

He is currently a Research Assistant Professor with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong. His research interests include integrity monitoring of GNSS, Kinematic GPS, sensor integration for various navigation systems.

Xuanyu Qu received his B.S. degrees and M.S. degrees from Chang'an University, Xi'an, China, in 2015 and 2018. He is currently pursuing the Ph.D. degree with the Research Institute of Land and Space, The Hong Kong Polytechnic University, Hong Kong.

His current research interests include precise GNSS positioning and structural health monitoring.



Weihao Ding received his B.S. and M.S. degrees in Petroleum Engineering from Texas A&M University, College Station, USA, in 2015 and 2017. He is currently pursuing the Ph.D. degree with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong.

His current research interests include PPP and GNSS in urban areas.

Wu Chen received his Ph.D. degree from Newcastle University, Newcastle upon Tyne, UK, in 1992.

He is currently a Professor with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong. His current research interests include the GNSS positioning quality evaluation, system integrity, various GNSS applications, seamless positioning and SLAM.