Hindawi
Journal of Sensors
Volume 2019, Article ID 3737265, 8 pages
https://doi.org/10.1155/2019/3737265

Research Article

Deep Learning-Based GNSS Network-Based Real-Time Kinematic Improvement for Autonomous Ground Vehicle Navigation

Hee-Un Kim and Tae-Suk Bae

Department of Geoinformation Engineering, Sejong University, Seoul 05006, Republic of Korea

Correspondence should be addressed to Tae-Suk Bae; baezae@sejong.ac.kr

Received 29 November 2018; Revised 13 February 2019; Accepted 4 March 2019; Published 31 March 2019

Guest Editor: Sang-Hoon Hong

Much navigation over the last several decades has been aided by the global navigation satellite system (GNSS). In addition, with the advent of the multi-GNSS era, more and more satellites are available for navigation purposes. However, the navigation is generally carried out by point positioning based on the pseudoranges. The real-time kinematic (RTK) and the advanced technology, namely, the network RTK (NRTK), were introduced for better positioning and navigation. Further improved navigation was also investigated by combining other sensors such as the inertial measurement unit (IMU). On the other hand, a deep learning technique has been recently evolving in many fields, including automatic navigation of the vehicles. This is because deep learning combines various sensors without complicated analytical modeling of each individual sensor. In this study, we structured the multilayer recurrent neural networks (RNN) to improve the accuracy and the stability of the GNSS absolute solutions for the autonomous vehicle navigation. Specifically, the long short-term memory (LSTM) is an especially useful algorithm for time series data such as navigation with moderate speed of platforms. From an experiment conducted in a testing area, the LSTM algorithm developed the positioning accuracy by about 40% compared to GNSS-only navigation without any external bias information. Once the bias is taken care of, the accuracy will significantly be improved up to 8 times better than the GNSS absolute positioning results. The bias terms of the solution need to be estimated within the model by optimizing the layers as well as the nodes each layer, which should be done in further research.

1. Introduction

In recent years, the autonomous navigating vehicle has been a most popular topic in the field of positioning. It is usually categorized as a vehicle that is navigating by introducing information & communication technology (ICT) to self-recognize the driving condition, make decisions, and finally control the route with minimal user intervention. Although there are many assistive systems for the driver such as Advanced Driver Assistance Systems (ADAS) or Collision Avoidance System (CAS), the full operational self-navigation is still not yet available.

The first autonomous navigation vehicle was initiated from the Grand Challenge in 2004 hosted by the Defense Advanced Research Projects Agency (DARPA) and afterwards held in a complex urban area [1]. As is well known, the global navigation satellite system (GNSS) has been playing an important role in ground vehicle navigation. Further improvement was achieved in positioning accuracy due to the introduction of correction information in the mid-1990s: differential GPS (DGPS) for pseudoranges and real-time kinematic (RTK) for carrier phases [2, 3]. However, the GNSS solution is vulnerable to signal blockage from such surrounding environments as tunnels and urban canyons, or unpredictable multipaths.

Although a decimeter or better accuracy is expected from RTK for short baselines, even more accurate positioning is possible in real time due to sophisticated correction methods, called the network RTK (NRTK) [4, 5]. However, this level of accuracy can only be attainable for the "static" processing, while the vehicle navigation is generally performed in a "kinematic" mode with tens of meters.

Autonomous navigation is generally on the premise of accurate positioning, but it cannot be guaranteed by GNSS

only. Many different types of sensors are combined to support accurate positioning, which includes RADAR or LiDAR for viewing which way to go, and inertial measurement unit (IMU) to track the linear and/or rotational movements of the vehicle. Therefore, it is necessary to integrate various sensors to make up for the weakness of GNSS and to estimate accurate position of the vehicles [6]. A detailed case study of autonomous car was discussed based on the distributed architecture [7]. In addition, numerous researches of integrating GNSS/IMU and RTK technique have been conducted for the applications of Unmanned Aerial Vehicles (UAVs), but most of them apply a single-frequency GNSS receiver [8, 9].

The filter to integrate sensor data requires an analytical model for each specific sensor. Currently, it includes GNSS, IMU, and onboard wheel speed sensor (WSS), but a great deal of sensors will be available in near future [10]. Therefore, the estimation filter needs to be redesigned to incorporate all those sensor data to integrate the position of the vehicle, resulting in extremely increased complexity of the filter.

The accurate and reliable position information for self-navigating vehicles is sometimes unachievable due to the vulnerability of the GNSS signal as mentioned above. The objective of this study is to develop a model to estimate the position of the vehicle without redesigning the analytical model of each individual sensor with constraints. We applied the deep learning technique to predict the position of the vehicle based on multisensor data including GNSS. The field experiment was carried out using the mobile mapping system (MMS) [11] with onboard sensors where the high-end GPS/INS system was used as a reference truth. The concept of deep learning is described in the next section, and the test results and discussion follow.

2. Framework of Deep Learning

Since the introduction of the neural events and the complicated logical means for nets [12], many researchers have been investigating how to mimic the human brain into the system. Although a perceptron algorithm based on supervised learning was already developed in mid-20th century [13], the deep learning technique was not applied in practice due to lack of computational resources for a long time. A new learning procedure, called back-propagation for networks, was suggested to repeatedly adjust the weights of the nodes in the network to minimize the difference between the actual output vector and the generated labeled data sets [14]. This back-propagation learning was applied to a practical problem of a handwritten zip code with short learning time and improved performance [15].

Most of the deep learning studies focus on the image-based applications such as remote sensing, image matching and classifications, and detection of road features by combining different sources [16–19]. The long short-term memory (LSTM) methods and a variance of recurrent neural networks (RNN) were mostly applied for the navigation of ground vehicles, UAVs, and robotics [20–22]. This is because the navigation data is provided in time series, which is suitable for the LSTM model. The MEMS IMU data is usually integrated with GNSS observations to complement the weakness of each sensor. Therefore, the deep learning technique can filter two sources of data without setting up the complicated analytical models [23].

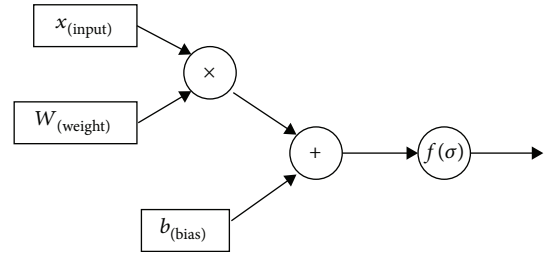

Figure 1: Basic framework of a single neuron.

The (artificial) neural network theory has tremendously impacted on many fields including the autonomous navigation of vehicles and many intelligent applications. Although the theory was suggested long ago, the neural network technique was only available in recent years due to the limitation of the computing resources as mentioned above.

The neural network is defined as a network of neurons, which imitates a structure of human neuron to make various decisions. The neurons transmit a signal if it exceeds a threshold value. For a simple model, we assume that a neuron receives signals from input neurons. It is necessary to consider three factors: the intensity of the input signal, the threshold value, and finally the intensity of the output signal. Once the input signals are received by the neurons, they are multiplied by the weight of each neuron, then combined into the output signal, which may or may not be transmitted depending on the threshold value. Figure 1 shows a basic framework of a neural network.

The process can be represented by the following equation:

$$\sigma = W \cdot x + b, \tag{1}$$

where $f(\sigma)$ represents the activation function that decides whether the neuron passes the value of σ or modifies it, and the term b represents the bias value to be added to the product of the input x times the weight W. Our goal is to estimate the weight of each neuron and the bias vector to better represent the output signal.

Figure 2 shows one of the activation functions, that is, a sigmoid function and its time derivative. The sigmoid function can be used to mimic a cumulative distribution function, and the output values are between 0 and 1, which is realistic in general cases. However, as we can see in Figure 2, the maximum value of the time derivative of the sigmoid function evaluated at 0 is about 0.25, which makes the system insensitive if we deal with the multiple hidden layers [24]. Therefore, in some cases, it is favorable to introduce different types of activation function such as Rectified Linear Unit (ReLU) [25]. The ultimate goal is to find the weight vector and bias by minimizing the cross-entropy function, that is, the amount of error under optimal condition. The

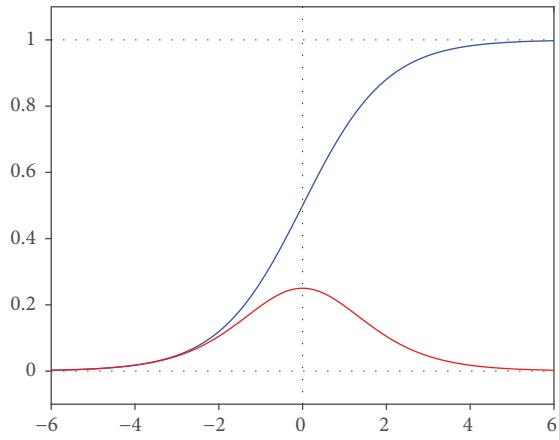

Figure 2: Example of the activation function also called a sigmoidal function, or just a sigmoid (blue) and its time derivative (red).

process can be done iteratively based on the gradient descent principle.

For linear nonseparable problems, the "multilayer" perceptron (MLP) was suggested to solve the nonlinear problem through the error propagation training process [14, 26]. The MLP algorithm is composed of input, hidden, and output layers, which can be described as follows:

$$h_n = f_n(W_{1n} \cdot x_1 + W_{2n} \cdot x_2 + \cdots + W_{mn} \cdot x_m + b_n),$$
$$y_n = g_n(W_n \cdot h_n + b_n), \quad (2)$$

where x_i is a vector variable for input data, W is the weight matrix, h_n is the output variable for the n^{th} hidden node, and f_n and g_n are the activation functions for input and output layers, respectively. The most important characteristic of MLP is its flexibility of input data types, resulting in varying number of hidden layers as well as number of nodes in the hidden layer depending on the input data. Since the weights of the internal hidden layers are adjusted during the process, the significance of each nodes is reflected, which is a useful feature of the back-propagation process [26].

A recurrent neural network (RNN) is an algorithm that is composed of a circulation structure [20, 22, 23, 27]. The hidden nodes have directional information; thus, past and current data are connected to constitute a circulation. Therefore, it is useful to analyze the time series data due to the inherent time information. The RNN algorithm has a flexible structure and diverse characteristics because the length of a sequence does not matter in this case. In addition, the RNN can be affected by the output of the previous data. Figure 3 shows the structure of the RNN framework.

Long short-term memory (LSTM) [20, 21, 23] is an expansion of the RNN algorithm by adding more gates to RNN hidden layers (see Figure 4) [28]. The gates are composed of three types, that is, input, output, and forget gates. One thing to be mentioned is that the forget gate has a role of determining whether to remove the past information or not depending on the output of activation function (ranges from 0 to 1).

3. Results and Discussion

3.1. Strategy of Experiment. As mentioned in Section 1, GNSS-only positioning cannot guarantee the required accuracy due to the availability of the signals in urban canyon or tunnels. For the experiment of the deep learning of ground vehicle navigation, we used the mobile mapping system (MMS) data to generate a model for estimating the vehicle tracks (see Figure 5). The experiment was conducted at the elongated shape of a parking lot in Daegu Gyeongbuk Institute of Science and Technology (DGIST) on November 9, 2015 (Figure 6).

The GPS/INS-integrated system, namely, POSLV, was mounted on the MMS vehicle, along with the additional independent GNSS receiver. The onboard sensors operate together through Controller Area Network (CAN) communication, which includes the wheel speed and turning speed sensors, the gravity sensor, the steering sensor, and the speedometer of the vehicle. The NovAtel GNSS receiver provides a navigation solution based on the pseudoranges. A more precise and accurate solution was calculated by the POSLV based on the carrier phase measurements and the integrated IMU sensors, which was used as a reference in this study. MTi-G-700 sensor [29] generates the accelerometer and gyroscope data, and the atmospheric pressure data are available as well. We used the yaw data at epoch $(t-1)$ to estimate the position of vehicle at epoch (t). Since the position and the velocity are basically resulted from the integration of the acceleration of the vehicle, the output of the accelerometer is tightly coupled with the position/velocity. However, for a simple implementation of deep learning technique, we trained the model based on the position from the GNSS absolute position (transformed into the planar coordinates), the wheel speed sensor data, and the yaw information obtained from the MTi-G-700 sensor. For more elaborate training of the model, the accelerometer data should be incorporated in further analysis. Table 1 summarizes the specification and the output of the sensors used in this study.

The open source library, Keras [16, 30], implemented using TensorFlow and Theano based on Python, was used for training and testing the data to evaluate the deep learning algorithm. In addition, since the modules in Keras are independently running, the user-defined model can be easily generated by combining the modules.

Figure 7 shows the graphical representation of the deep learning testing scheme used in this study. Two thirds of data were used to train the model, and the remaining was reserved for evaluating the performance and accuracy of the model. The raw data includes GNSS, IMU, and from the onboard vehicle sensor. The LSTM model was used in this study, which is specifically useful for the time series data as mentioned above. It is a supervised learning, and the model is optimized by the training process. Finally, the predicted position of the vehicle was compared with the reference position to calculate the accuracy of the model.

3.2. Analysis of the Results. Figure 8 shows the trajectory of the vehicle where two solutions are plotted together, one for

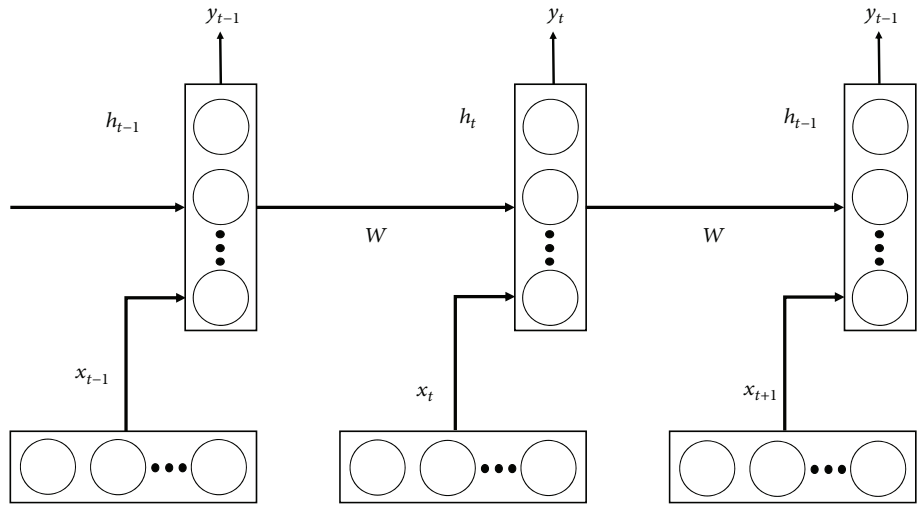

Figure 3: Structure of recurrent neural networks (RNN). The circle and the rectangle represent the node and the layer in the process, respectively.

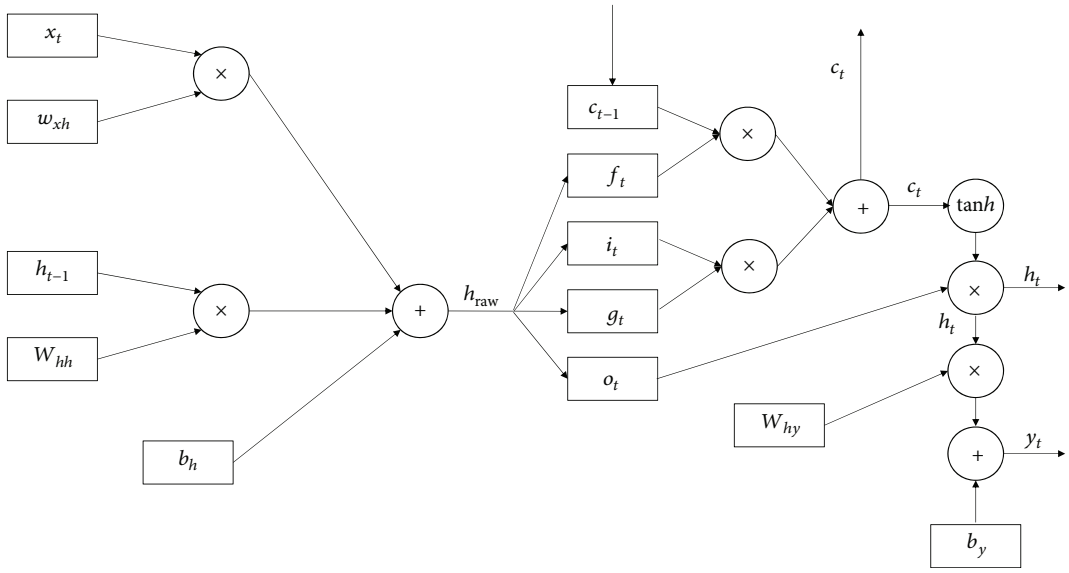

Figure 4: The schematic diagram of LSTM [28].

Figure 5: The MMS vehicle to collect data in this study (courtesy of Dr. J. H. Han).

the postprocessed POSLV solution and the other which is the absolute positioning based on pseudoranges. Both solutions represent the absolute coordinates of the vehicle in a global reference frame. The difference between solutions at each epoch is around 3.5 m (horizontal accuracy) in terms of Root Mean Squared Error (RMSE), while the North-South component takes most part of the errors (see Table 2). The antenna of the GNSS receiver is installed in the front, and the POSLV for the reference solution is equipped in the rear of the vehicle, resulting in an (lever arm) offset between two reference points. Since the shape of the parking lot for data acquisition is elongated in the N-S direction, it seems that the offset immediately affects the mean error in that direction. Therefore, the offset needs to be calibrated beforehand, and furthermore, it is necessary to estimate the amount of

Figure 6: Test bed of this study (parking lot of DGIST).

bias in the GNSS solution during the deep learning training process, which should be investigated further.

As discussed above, the LSTM model generally performs better than any other model, especially for the time series data. However, the (assumed) input data from NovAtel pseudorange solution is clearly biased from the reference data as seen in the trajectory plot. The interesting point is that the "errors" have an almost circular shape as can be seen in Figure 9, even though the testing site is elongated in the North-South direction.

Even in the LSTM model, the bias can affect the final predicted solution. Therefore, the bias term as well as the circular deviation was estimated using the conventional Least-squares Solution (LESS) to isolate the systematic errors from the solution based on the deep learning technique. The biases are estimated as 2.54 m and -0.95 m for North and East components, respectively, along with the radius of 2.73 m. Once the bias is corrected, the RMSE drops down to 0.83 m in the horizontal plane, although most of the errors are attributed to the North-South component.

The LSTM algorithm is a recurrent neural network in which the data in the past can affect the current data. Therefore, it generally performs well for the vehicle navigation, which is basically a time series data. Through several experiments, we set up the training model with two hidden layers with 4 and 2 nodes each. In addition, the mean squared error was used to calculate the loss function, while the Adam was used for the activation function. The input data is composed of 4 features, that is, two horizontal components, the speed and the yaw information.

3.3. Prediction for Vehicle Navigation. For the vehicle navigation based on deep learning, we prepared the input data from the sensor output. The GNSS sensor (NovAtel DL-V3-GENERIC) provided an absolute positioning solution in the Cartesian coordinate system of a global frame at the 1 Hz interval. However, the IMU and onboard sensors have higher rates at 100 Hz, thus the gaps of GNSS output should be filled up to match the time interval. For the training stage, the GNSS solution was interpolated at a 100 Hz interval using the (future) output solution (corresponds to Interpol. in Table 2). However, since it is not available for interpolation in real-time application, we generated an extrapolated position based on the speed of the vehicle from onboard sensors and the previous positions (corresponds to Extrapol. in Table 2). Both interpolated and extrapolated cases are compared with the reference solution, that is, POSLV solution, to analyze the solution characteristics after transforming into the horizontal frame (North-East-Up local reference frame).

Table 3 shows the statistics of the prediction results for the horizontal component by the LSTM model. The original GNSS absolute positioning result shows an RMSE of about 3.8 m, but it drops down to 2.3 m with extrapolated position based on the previous results. There is almost 40% of improvement in the predicted horizontal coordinates (North-East frame given in unit of meters) once the interpolated position is available. Interestingly, there is no significant difference between interpolation and extrapolation in terms of positioning accuracy. Therefore, it can certainly be applied to real-time applications.

As mentioned above, there is a systematic bias in the GNSS absolute solution due to the (lever arm) offset. Once this bias is removed from the input position information, the magnitude of the bias is clearly decreased, and almost more than 80% of improvement was observed. The RMSE of planar coordinates is about 0.45 m in real situations, and the interpolated and extrapolated solutions show a similar performance in all statistics. Therefore, instead of estimating the bias terms in a separate process as done in this study, it may further improve the results if they are incorporated in the deep learning model as a system.

The most critical issue in the prediction of vehicle location is the accuracy and reliability of the solution. The GNSS signal is sometimes blocked by the surrounding environment, for example, skyscrapers in urban areas or tunnels. Therefore, the problem may be solved by establishing a model to combine the IMU data as well as onboard sensors.

Figure 10 shows the loss and the accuracy of the LSTM model at each epoch to check the training and the test processes. As can be seen in the figure, the loss function quickly drops down and converges to zero, and conversely the accuracy gets closer to 1.0 and continues to stay. Therefore, it can be concluded that the learning process of the model was conducted properly.

3.4. Discussion of the Results. As a preliminary study to apply deep learning technique for the positioning of ground vehicle, the conventional GNSS absolute solution was used as an input for the training purpose. However, the GNSS receiver was installed separately from the POSLV system; thus, there is an inevitable offset in the comparison result. This virtual bias, caused by a lever arm, was absorbed by the deep learning technique in this study (about 40%), regardless of interpolation or extrapolation of the speed of the vehicle. Once the bias is removed before training a model, only several decimeters of error were achieved, which has

Table 1: Specification of sensors used in this study.

	GNSS	IMU	OBD	GNSS/INS*
Sensor	NovAtel DL-V3	MTi-G-700	OBD-II	POSLV520
Specification	Dual-frequency GPS/GLONASS	Roll/pitch (0.3°, dynamic), yaw (1.0°)	Wheel speed, Steering angle	Dual frequency GPS/GLONASS, Roll/pitch (0.005°), Yaw (0.015°)
Frequency	1 Hz	100 Hz	100 Hz	100 Hz
Output	XYZ (m)	Yaw (deg)	Speed (km/h)	XYZ (m)

The symbol ∗ represents the reference truth for comparison of the results.

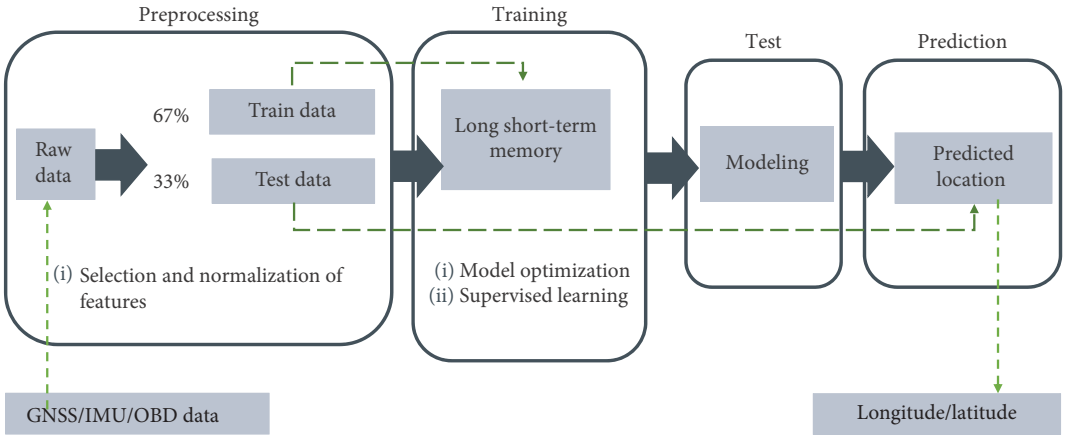

Figure 7: Overall scheme of the deep learning process used in this study [31].

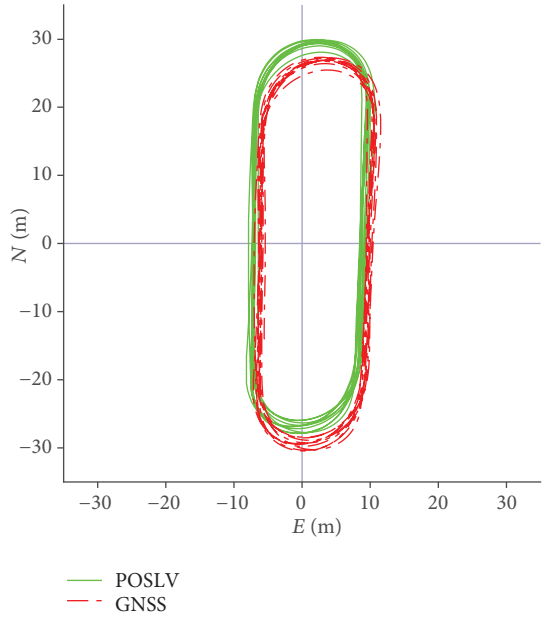

Figure 8: Trajectory of the test vehicles in the parking lot where the absolute positioning results are dotted in red and the postprocessed POSLV in green was used as a reference.

significant importance for the autonomous ground vehicle navigation.

One thing to be mentioned is that the experiment of this study was done in a limited environment, that is, a small area parking lot. Therefore, there are many factors to consider the application of real road conditions, which includes varying speed, rapid and/or sharp turns, and in unfavorable cases the loss of GNSS signals due to the surrounding environment. For general application, we need to secure enormous data for training where the acceleration and the rotational information from IMU data will play an important role to reduce time and upgrade the performance.

Table 2: Statistics of the absolute positioning solution compared to the reference POSLV solution.

	North	East	Horizontal
Mean (m)	2.53	-0.95	3.33
Std. dev. (m)	2.46	1.16	1.91
RMSE (m)	3.53	1.49	3.83

The LSTM model was applied for the time series data of vehicle navigation to reflect the information in the past. We adopted only two hidden layers with 4 and 2 nodes for simplicity although numerous trainings were applied to find better results. The number of layers and nodes in each layer should also be optimized, which is still an open question in deep learning technique.

4. Conclusion

While most researches on deep learning technique were focused on image-based applications, we proposed an integrated navigation algorithm with various sensors (GNSS and onboard sensors) in this study. Since the GNSS itself

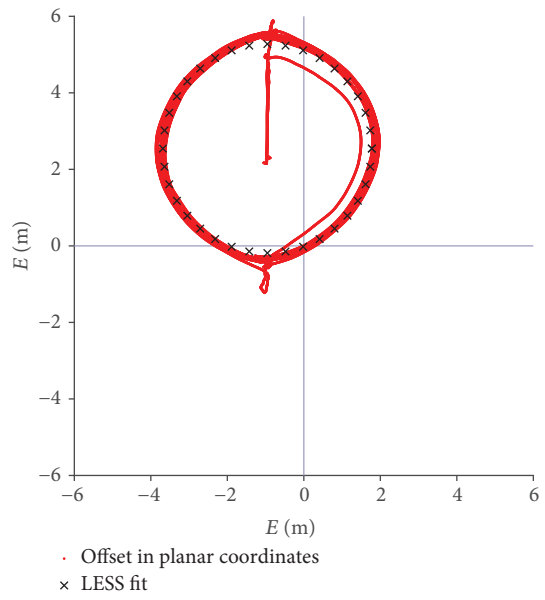

Figure 9: Offset of the NovAtel absolute solution from the reference truth in a planar coordinate system, along with the estimated LESS to remove the bias.

Table 3: The statistics of horizontal positioning accuracy of the LSTM model with/without bias handling.

	Abs. solution	With bias		Bias removed	
		Extrapol.	Interpol.	Extrapol.	Interpol.
Mean (m)	3.33	2.22	2.22	0.37	0.37
Std. dev. (m)	1.91	0.60	0.57	0.25	0.27
RMSE (m)	3.83	2.30	2.29	0.45	0.46

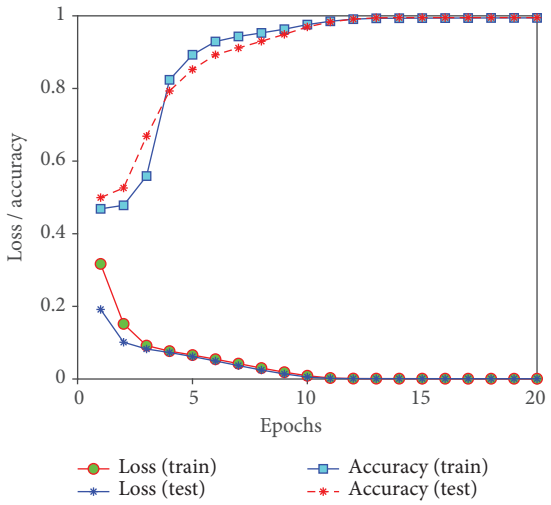

Figure 10: The loss and the accuracy of the LSTM model during training and testing processes.

does not provide sufficiently accurate positioning information, additional information should be provided for the autonomous navigation of vehicles. The deep learning algorithm was studied to test the performance and the accuracy of the positioning based on neural network theory. Instead of setting up a complex mathematical model for the measurements, the deep learning technique was applied to predict the vehicle location to improve the accuracy and the expandability. Since the navigation data is mostly given in time series, the LSTM algorithm was applied.

The GNSS sensor only provides the RMSE of about 3.8 m, which is the absolute positioning result based on the pseudorange data. When the GNSS position was interpolated at 100 Hz, its output was improved to about 2.3 m with the LSTM model (but only available in a postprocessing mode). Even in a real-time application with the extrapolated position, a comparable accuracy was obtained with no prior bias information. The MMS used in this study has a possible bias due to the lever arm of the GNSS sensor. The bias was removed using the conventional LESS approach. Once the bias was removed from the input data, the LSTM model results were significantly improved to about 0.45 m, which is almost identical for both cases of interpolation and extrapolation.

In this study, a complex mathematical model was not set up for each individual sensor, which gives high flexibility in the integration of different kinds of sensors for practical application of future autonomous ground vehicle navigation. Although much progress was shown in this study, it still needs to be elaborated to incorporate sensors with more features as input data and the strategy on how to balance the weight between sensor data. Since the current experiment is slightly limited in the amount as well as the pattern of data (for example, data acquisition on the same trajectory), it is necessary to conduct the experiment in real road situations with enormous training data. In addition, the bias information should also be integrated into the model to be estimated during the training process, which is anticipated to generate a more practical prediction model for the autonomous vehicle navigation.

Data Availability

All data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A1B07048475).

References

[1] M. Buehler, K. Iagnemma, and S. Singh, "The 2005 DARPA grand challenge – the great robot race," *Industrial Robot: An International Journal*, vol. 35, no. 6, 2008.

[2] G. R. Hu, V. H. S. Khoo, P. C. Goh, and C. L. Law, "Internet-based GPS VRS RTK positioning with a multiple reference station network," *Journal of Global Positioning Systems*, vol. 1, no. 2, pp. 113–120, 2002.

[3] E. Umnig, G. Möller, F. Hinterberger, and R. Weber, "GNSS RTK-networks: the significance and issues to realize a recent reference coordinate system," in *EGU General Assembly 2014*, Vienna, Austria, April–May 2014.

[4] V. Janssen, "A comparison of the VRS and MAC principles for network RTK," in *IGNSS Symposium 2009*, Qld, Australia, December 2009.

[5] C. Wang, Y. Feng, M. Higgins, and B. Cowie, "Assessment of commercial network RTK user positioning performance over long inter-station distances," *Journal of Global Positioning Systems*, vol. 9, no. 1, pp. 78–89, 2010.

[6] S. Rezaei and R. Sengupta, "Kalman filter-based integration of DGPS and vehicle sensors for localization," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 6, pp. 1080–1088, 2007.

[7] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car—part II: a case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 5119–5132, 2015.

[8] M. Strohmeier and S. Montenegro, "Coupled GPS/MEMS IMU attitude determination of small UAVs with COTS," *Electronics*, vol. 6, no. 1, p. 15, 2017.

[9] T. Li, H. Zhang, Z. Gao, Q. Chen, and X. Niu, "High-accuracy positioning in urban environments using single-frequency multi-GNSS RTK/MEMS-IMU integration," *Remote Sensing*, vol. 10, no. 2, p. 205, 2018.

[10] D. Yuan, L. Cai, M. Li, C. Liang, and X. Hou, "Multi-sensor integration based on a new quantum neural network model for land-vehicle navigation," *NeuroQuantology*, vol. 16, no. 6, pp. 619–624, 2018.

[11] S. Blaser, S. Nebiker, and S. Cavegn, "System design, calibration and performance analysis of a novel 360° stereo panoramic mobile mapping system," in *2017 ISPRS*, pp. 207–213, Hannover, Germany, June 2017.

[12] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[13] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[15] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[16] J. E. Ball, D. T. Anderson, and C. S. Chan, "Comprehensive survey of deep learning in remote sensing: theories, tools and challenges for the community," *Journal of Applied Remote Sensing*, vol. 11, no. 4, pp. 1–64, 2017.

[17] C. Fernández, D. Fernández-Llorca, and M. A. Sotelo, "A hybrid vision-map method for urban road detection," *Journal of Advanced Transportation*, vol. 2017, Article ID 7090549, 21 pages, 2017.

[18] J. Li, H. Bao, X. Han et al., "Real-time self-driving car navigation and obstacle avoidance using mobile 3D laser scanner and GNSS," *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 23017–23039, 2017.

[19] D. G. Lee, E. J. Cho, and D. C. Lee, "Evaluation of building detection from aerial images using region-based convolutional neural network for deep learning," *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, vol. 36, no. 6, pp. 469–481, 2018.

[20] C. Jiang, S. Chen, Y. Chen et al., "A MEMS IMU de-noising method using long short term memory recurrent neural networks (LSTM-RNN)," *Sensors*, vol. 18, no. 10, 2018.

[21] F. Altché and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, Japan, October 2017.

[22] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, "A review of deep learning methods and applications for unmanned aerial vehicles," *Journal of Sensors*, vol. 2017, Article ID 3296874, 13 pages, 2017.

[23] S. Hosseinyalamdary, "Deep Kalman filter: simultaneous multi-sensor integration and modelling; a GNSS/IMU case study," *Sensors*, vol. 18, no. 5, article 1316, 2018.

[24] Y. Sugomori, *Detailed Understanding: Time Series Data by Tensorflow · Keras*, Minavi Publishing, 2017.

[25] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[26] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 296–298, 1990.

[27] C. Olah, *Understanding LSTM Networks*, Colah's Blog, 2018, http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] Xsens, *MTi-G-700 3D Motion Tracking Sensor*, 2018, Xsens products homepage, https://www.xsens.com/products/mti-g-710/.

[30] Keras, "Keras: The Python deep learning library," 2018, https://keras.io.

[31] H. Kim and T. S. Bae, "Preliminary study of deep learning-based precipitation prediction," *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, vol. 35, no. 5, pp. 423–430, 2017.