sensors

MDPI

Article

A Novel KGP Algorithm for Improving INS/GPS Integrated Navigation Positioning Accuracy

Huibing Zhang [1], Tong Li [1], Lihua Yin [2,*], Dingke Liu [1], Ya Zhou [1], Jingwei Zhang [1] and Fang Pan [3]

[1] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China; zhanghuibing@guet.edu.cn (H.Z.); 1603303018@mails.guet.edu.cn (T.L.); liudingke00@163.com (D.L.); ccyzhou@guet.edu.cn (Y.Z.); gtzjw@guet.edu.cn (J.Z.)

[2] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China

[3] Teaching Affairs Office, Guangxi Normal University, Guilin 541004, China; panfang@mailbox.gxnu.edu.cn

* Correspondence: yinlh_iie@163.com

Received: 21 February 2019; Accepted: 1 April 2019; Published: 4 April 2019

Abstract: The fusion of multi-source sensor data is an effective method for improving the accuracy of vehicle navigation. The generalization abilities of neural-network-based inertial devices and GPS integrated navigation systems weaken as the nonlinearity in the system increases, resulting in decreased positioning accuracy. Therefore, a KF-GDBT-PSO (Kalman Filter-Gradient Boosting Decision Tree-Particle Swarm Optimization, KGP) data fusion method was proposed in this work. This method establishes an Inertial Navigation System (INS) error compensation model by integrating Kalman Filter (KF) and Gradient Boosting Decision Tree (GBDT). To improve the prediction accuracy of the GBDT, we optimized the learning algorithm and the fitness parameter using Particle Swarm Optimization (PSO). When the GPS signal was stable, the KGP method was used to solve the nonlinearity issue between the vehicle feature and positioning data. When the GPS signal was unstable, the training model was used to correct the positioning error for the INS, thereby improving the positioning accuracy and continuity. The experimental results show that our method increased the positioning accuracy by 28.20–59.89% compared with the multi-layer perceptual neural network and random forest regression.

Keywords: INS/GPS integrated navigation; data fusion; Gradient Boosting Decision Tree

1. Introduction

With the development of sensor technology, context-aware vehicles (e.g., location services and automatic driving) are becoming increasingly popular. However, these applications require high context perception accuracy, especially in assisted and automatic driving, which has increasingly high requirements for the continuity, reliability, and accuracy of vehicle positioning. The positioning performance of a single Global Positioning System (GPS) may be decreased by various factors, such as occlusion and interference when driving on urban roads [1,2]. Thus, it is difficult to meet the needs of the Internet of Vehicles. However, multi-source sensor fusion, e.g., Inertial Navigation System (INS) and GPS integrated navigation systems, can effectively solve these problems [3,4].

For the data fusion problem in integrated navigation, Kalman Filter (KF) is the existing optimal trajectory estimation method, which solves the problem of tedious calculation caused by the weak nonlinear ability. Particle Filter (PF) is considered a benchmark of the filtering method for predicting vehicle position, but the large number of particles required by PF leads the algorithm computationally expensive. In integrated navigation, when the GPS signal is interrupted [5,6], the positioning error in the Inertial Measurement Unit (IMU) accumulates over time, decreasing the overall performance of the integrated navigation [7–9]. To improve the positioning performance during GPS signal interruption, Artificial Neural Networks (ANNs) have been introduced into the INS/GPS integrated navigation system, e.g., Multilayer Perceptron Neural Networks (MLPNNs) [10,11], Radial Basis Function Neural Networks (RBFNNs) [12–14], Long Short Term Memory Recurrent Neural (LSTM-RNNs) [15], networks and adaptive Neuron-Fuzzy Inference Systems (ANFISs) [16,17]. The main idea is to train the relationship between the vehicle feature data and INS errors through ANNs when the GPS signal is available. When the GPS signals are available, the current or latest training model is used to predict the positioning data. This method effectively reduces the positioning error and ensures positioning continuity. Since the ANN is trained completely using input data, its generalization ability is limited when the vehicle state data during training is different from that during prediction. To solve this problem, in one study [18], ensemble learning was included in the INS/GPS integrated navigation system, which effectively improved the generalization ability. Although the Least Squares Boosting and Bagging algorithms proposed in that study [18] could improve the positioning accuracy, the errors of the INS internal sensor (e.g., steering deviation, running deviation, and scale factor drift) increased the nonlinear complexity of the relationship between the input and output data. The model was weak at recognizing feature variables, resulting in unsatisfactory predictions of positioning. In addition, the sensor's errors accumulated over time. When the GPS signal loss was over 5 min, the prediction accuracy of the ensemble learning scheme began to decrease gradually [18].

To solve the above-mentioned problems, a Kalman Filter-Gradient Boosting Decision Tree-Particle Swarm Optimization (KF-GBDT-PSO, henceforth denoted KGP) data fusion method is proposed herein [19], which consists of two consecutive phases: Training and prediction. In the training phase, the KGP prediction model can compensate for the INS positioning error through the relationship between vehicle feature data and KF estimations of the positioning error [20,21]. In the prediction phase (during GPS signal loss), the trained model immediately predicts the positioning data. Compared with ANNs, the predicted values of the GBDT are obtained through accumulating the residuals of multiple trees. Due to its advantage of reducing model deviations, the Gradient Boosting Decision Tree (GBDT) provides a better generalization ability with better accuracy. Additionally, selecting regression parameters can be challenging. Thus, Particle Swarm Optimization (PSO) is introduced in the training phrase to select the optimal parameters for GBDT [22]. The KGP not only could extract nonlinear vehicle feature data in parking and driving states using the addition model and the forward distribution algorithm but could also use the Huber loss function to eliminate the location outliers collected due to road complexity. By flexibly covering various types of variables, the error rate of the integrated navigation system was effectively reduced [23], increasing the prediction accuracy of the positioning.

The remainder of this paper is organized as follows. In Section 2, an overview of the GBDT and PSO theories is provided. In Section 3, the integration scheme for the KGP method is introduced and discussed in detail. The experimental results are discussed in Section 4. Conclusions are presented in Section 5.

2. Improved Methods

2.1. GBDT Regression Algorithm

The GBDT consists of a gradient boosting and regression decision tree. The decision tree uses a Classification and Regression Tree (CART) as the base learner, which makes predictions quickly but can easily overfit [24]. Gradient boosting can improve the model performance and reduce the fitting ability of the decision tree by changing the weight of the sample [19]. The gradient boosting method and the decision tree learning algorithm complement each other, increasing the overall performance.

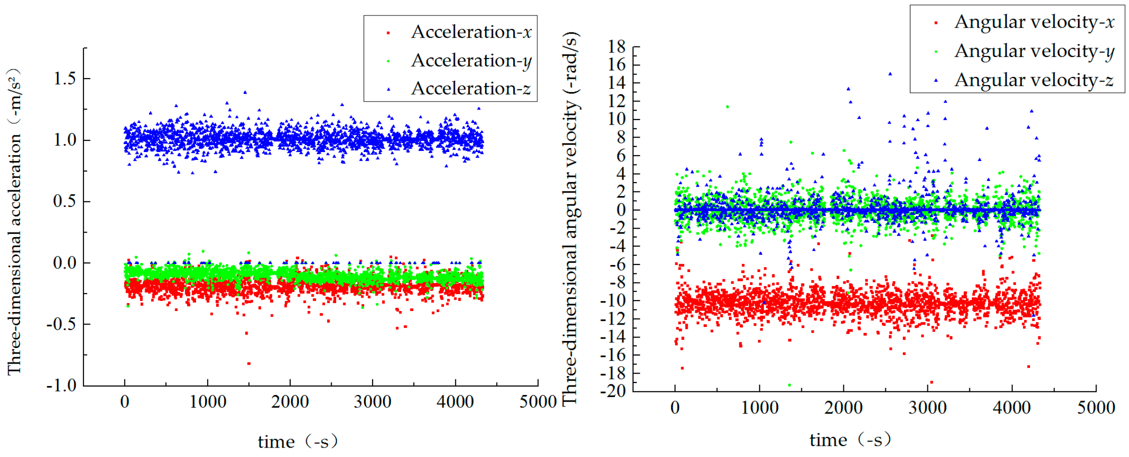
When training the model, we input the training sample $D = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i is the feature vector of the ith input sample, y_i is the sample tag of x_i, and n is the number of sample feature. Next, we initialized the learner as follows:

$$f_0(x) = arg \min_{c} \sum_{i=1}^{N} L(y_i, c). \tag{1}$$

We used a strong learner $f(x_i)$ and sample tag y_i to construct the loss function $L(y_i, f(x_i))$. Freidman [19] defined the loss function as:

$$L(y_i, f(x_i)) = \sum_{i=1}^{N} \frac{1}{2}[y_i - f(x_i)]^2. \tag{2}$$

By observing the data characteristics of the vehicle, we found that the gyroscope and accelerometer contained in the IMU are in the process of measuring the real road, and the data outliers will inevitably be collected due to the interference of the urban environment. In order to visually observe the measured values, the data in the three directions of acceleration and angular velocity are displayed by a scatter plot (Figure 1); it is well known that the anomaly measurement information increases the complexity of the system and largely affects the prediction accuracy of the navigation model.



(**a**) (**b**)

Figure 1. Scatter plot of reference data: (**a**) acceleration and (**b**) angular velocity.

Therefore, in this study, the loss function is defined as follows based on real data:

$$L(y_i, f(x_i)) = \begin{cases} \frac{1}{2}[y_i - f(x_i)]^2 & |y_i - f(x_i)| < \delta \\ \delta|y_i - f(x_i)| - \frac{\delta}{2} & |y_i - f(x_i)| < \delta. \end{cases} \tag{3}$$

Here, δ is measured by quantile, and the value is 0.9. The Huber loss function uses absolute loss for anomaly vehicle data. For normal data, we used the mean square error to effectively exclude inappropriate data. This not only maintains the continuity of the loss function, but also has better robustness to outliers.

In order to ensure the continuous decline of the loss function, the negative gradient of the loss function is used in the iteration to calculate the approximate value of the current model residuals. For the m^{th} iteration, the negative gradient is defined as:

$$r_{im}(y_i, f(x_i)) = \begin{cases} y_i - f(x_i) & |y_i - f(x_i)| \leq \delta \\ \delta sign(y_i - f(x_i)) & |y_i - f(x_i)| \geq \delta. \end{cases} \tag{4}$$

We fit (x_i, r_{im}) to obtain t CART regression trees. Each tree is represented by $h_t(x)$, and its corresponding leaf node area is $\{R_i\}_1^J$, where J represents the number of leaf nodes in the regression tree. The linear search is used to estimate values of each leaf node region of the regressed trees, which can minimize the loss function.

The traditional GBDT model does not introduce regularization, so it is easy to overfit complex data. To prevent this phenomenon, a regularization term (5) is introduced in the loss function to penalize the number of leaf nodes in each CART tree, which is equivalent to pruning the regression tree during the training process.

$$\Omega f = \gamma T + \frac{1}{2}\lambda \|\omega\| \tag{5}$$

Initially, the learning weights of the samples were the same. As the regression tree grew with iterations completed, samples' weights were updated. Samples having low prediction accuracies were assigned larger weights, and high accuracy samples were assigned smaller weights. The weights were defined as follows:

$$\rho_{\mathrm{m}} = \underset{\rho}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \rho_{\mathrm{m}} \mathrm{h}_m(x)). \tag{6}$$

Through updating the residuals of the regression trees, we obtained a strong learner model as follows:

$$f_m x_i = f_{m-1}(x_i) + \rho_{\mathrm{m}} \mathrm{h}_m(x). \tag{7}$$

Therefore, the final GBDT model is

$$F(x) = F_0(x) + v \sum_{i=1}^{N} \rho_{\mathrm{m}} \mathrm{h}_m(x), \tag{8}$$

where $v(0 < v < 1)$ is the learning rate that determines the iteration of GBDT.

2.2. PSO Algorithm

The regression parameters in the GBDT determine the prediction accuracy of the model. Parameter selection requires extensive experience or large-scale searching. We introduced PSO into the GBDT to search for high-quality parameters [25], as it was easy to implement and improved the global optimization and convergence velocity.

In the PSO algorithm, each particle represents a parameter. The model's particle dimension is $\theta = (v, m, l, d)$, where v, m, l, d respectively represent the learning rate, the number of iterations, the minimum number of leaves, and the maximum depth of the regression tree in the GBDT.

In four-dimensional space, we generate a set of position vectors X_θ and flight velocity vectors V_θ. Each particle represents an adaptive value assigned by the objective function $fitness\ (\cdot)$. To obtain the optimal objective values of the regression parameters, the particles' positions and velocity s are updated with reference to their two current extreme values as follows [26]:

$$V_\theta(t) = \omega V_\theta(t-1) + c_1 r_1 \left(P_{best}^{\theta} - X_\theta(t-1)\right) + c_2 r_2 \left(G_{best}^{\theta} - X_\theta(t-1)\right), \tag{9}$$

$$X_\theta(t) = X_\theta(t-1) + V_\theta(t). \tag{10}$$

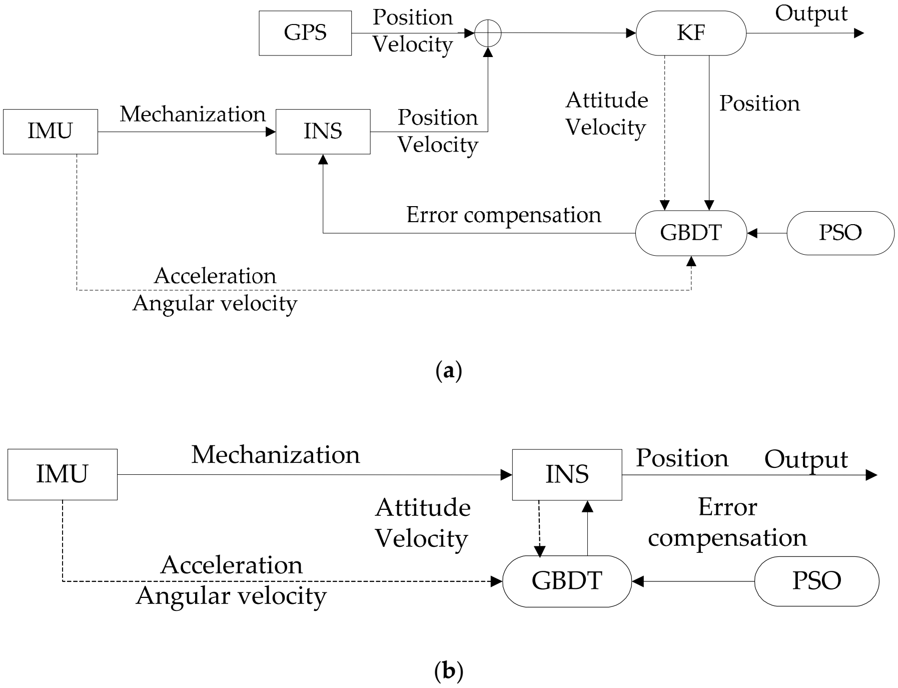
where ω is a negative inertia factor; c_1 and c_2 are the particle learning rate and global learning rate, respectively; r_1 and r_2 are random numbers between 0 and 1; and P_{best}^i and G_{best}^i represent the particle and global best locations, respectively.

The algorithm continuously updates P_{best}^i and G_{best}^i based on the calculated particle position, velocity, and adaptive function values. The regression parameter's optimal value is determined when the maximum number of iterations or the accuracy requirement is reached.

3. Integration Scheme of KGP

An "East-North-Up" (ENU) geographic coordinate system was used in this work as the vehicle navigation coordinate system. The system's origin is located at the vehicle's center of mass, where the x and y axes are in tangential directions of the local meridian and parallel, respectively, and the z axis is in the vertical direction.

Most vehicles run closely to the ground; the horizontal plane error was used in this study as an indicator to measure the performance of the vehicle navigation system during GPS signal interruptions. Figure 2 describes the INS/GPS integrated navigation system based on KGP.



(**a**)

(**b**)

Figure 2. KGP prediction model diagram: (**a**) training mode and (**b**) prediction mode.

The KGP includes two phases: training and prediction. We input vehicle data in the east-north direction (including velocity, posture, and the outputs of the gyroscope and accelerometer) to the GBDT model. The GBDT model output estimated the two-dimensional position errors of the KF filter. The KGP predictive model's input/output formulas are defined as follows:

$$\text{Input}: \quad X(t-s) = \left[V^e_{t-s}\ V^n_{t-s}\ \Psi^e_{t-s}\ \Psi^n_{t-s}\ w_{t-s}\ a_{t-s}\right]. \tag{11}$$

$$\text{Output}: \quad \Delta Y_t = \left[\Delta p^e_t\ \Delta p^n_t\right] = \left[p^e_t - p^e_{t-s}\ p^n_t - p^n_{t-s}\right]. \tag{12}$$

The subscript s indicates the data processing time interval; $[V^e_{t-s}\ V^n_{t-s}]$ and $[\Psi^e_{t-s}\ \Psi^n_{t-s}]$ are the velocity and angles in the east-north direction, respectively; w and a respectively represent the angular velocity from the gyroscope and the acceleration from the accelerometer; and p^e_t and p^n_t respectively represent the east and north positions after the latitude and longitude conversion.

When the GPS signal is normal, the system is in training mode; its function diagram is shown in Figure 2a. The system builds the KGP model using relationships between the temporary input/output variables:

$$P(x) = P_0(x) + v \sum_{i=1}^{N} \rho_t h_t(x). \tag{13}$$

The above formula is used to calculate vehicle data, such as the velocity, posture, and relationship between the IMU output and the KF estimation error through fitting.

To improve the quality of the GBDT regression parameters, PSO was adopted. The c_1, c_2, and ω parameter values are shown in Table 1 [27]. The adaptive function was defined as the standard mean squared error (MSE):

$$Fit(t) = \frac{1}{N}\sum_{d=1}^{N}(y_s - \hat{y}_d)^2. \quad (14)$$

In the above equation, $Fit(t)$ represents the fitness value of the t-dimensional particle, y_s is the sample output value, and $\hat{y}_d$ is the sample prediction value. The optimal value can be obtained by continuously iterating the GBDT regression parameters. The values are shown in Table 2.

Table 1. Parameters for PSO.

Parameter	Value
Particle factor (c_1)	0.103
Population factor (c_2)	2.897
Inertia weight (ω)	0.6

Table 2. Values of parameters for GBDT.

Parameter	Value
Learning rate (ν)	0.05
Number of iterations (m)	514
Minimum number of leaf (l)	2
Max depth(d)	8

When the GPS signal is interrupted, the KGP switches to prediction mode, as shown in Figure 2b. At this time, only the INS is operating in the integrated navigation system. The system sends the combined prediction estimation error back to the INS and generates predicted positioning data while correcting the INS position error in real time.

4. Results and Discussion

Two experiments were conducted. In Experiment 1, the generalization ability of the training model to the INS error under different driving states of the vehicle was investigated. In Experiment 2, the KGP model's ability to compensate for the INS error in its positioning predictions was tested. We also compared the results from the KGP with that from the MLPNN and random forest regression (RFR) [28,29].

We used the Chery eQ model to build the experimental platform, with the built-in GNSS receiver model MC20 and MG10 inertial navigation system as the hardware measurement equipment to collect reference data. The GPS position measurement accuracy was less than 2 m; the output frequency was 1 Hz; and the velocity measurement accuracy was 0.185 m/s, updated at 1 Hz. The output frequency of the INS inertial measurement unit was 1 Hz, and the sample IMU was used to acquire the linear and angular velocity of the vehicle at 1 Hz. The study was conducted in Guilin City, Guangxi Province, China, under five different road conditions: Straight, curved, sloped, downhill, and at an intersection with acceleration and deceleration.

4.1. Model Generalization Ability

The vehicle was tested on the road, shown in Figure 3. The tests included three types: A normal driving test (e.g., stops at traffic lights, corner turns, acceleration, and deceleration), a parking test (with only engine working), and a combined test (normal driving and subsequent parking). Figure 4 shows the vehicle's velocity s during the three tests over 48 min. Table 3 lists the time frames for the tests.

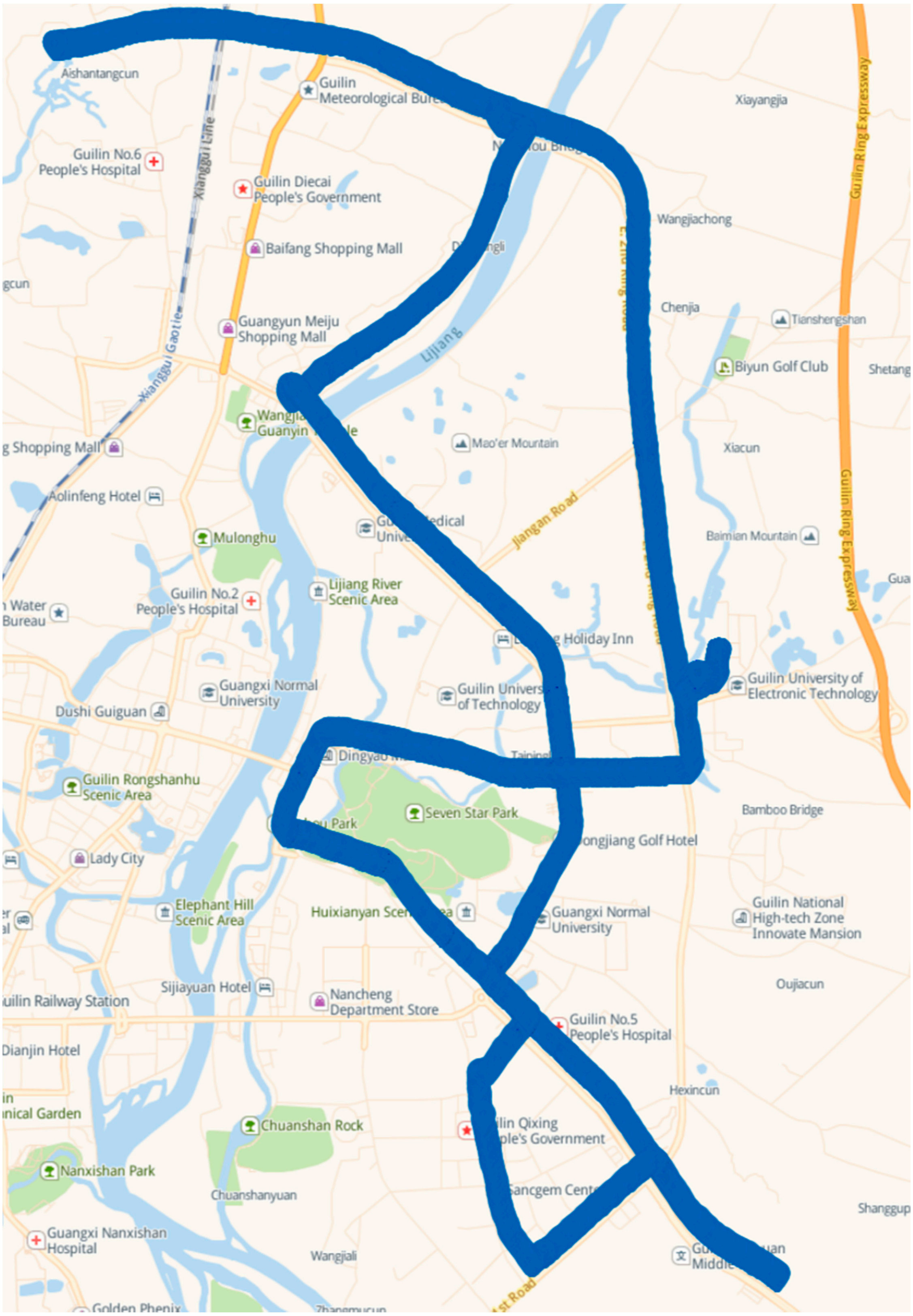


Figure 3. Experiment 1: Vehicle driving trajectory.

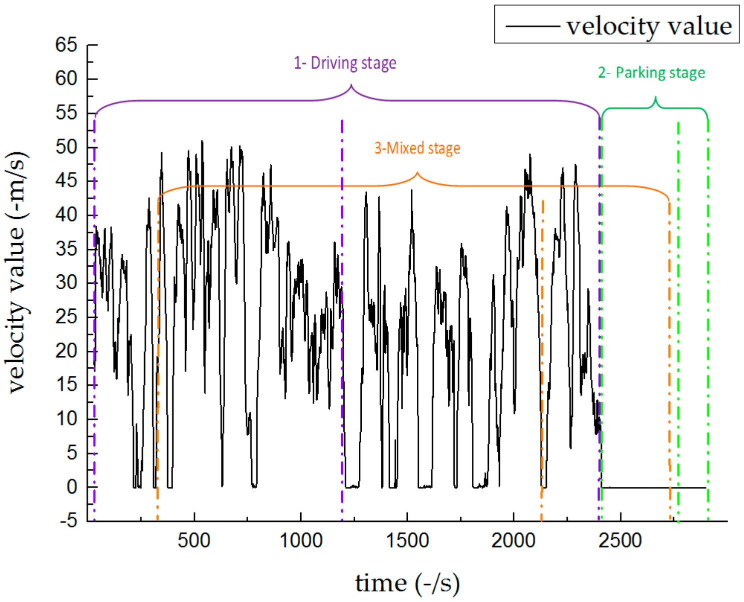


Figure 4. Experiment 1: Vehicle driving velocity.

Table 3. Time allocations of different independent stage.

	GPS On (s)	GPS Loss(s)	Total Time (s)
Driving	29–1192	1192–2429	2400
Parking	2429–2771	2771–2909	480
Combined	328–2129	2129–2728	2400

4.1.1. Results of the Normal Driving Test

Figure 5 depicts three algorithms that predicted the positioning error of the INS in the east-north direction. The RFR algorithm obtained better predictions [22] in shorter times. The prediction error of the RFR gradually decreased after the GPS signal was interrupted for over 5 min. Since we use the squared difference as the loss function, the MLPNN algorithm yielded better predictions in the latter half. The KGP algorithm was better than other algorithms in that it maintained good generalization ability for 20 min, reduced the positioning error in the north direction from 142 to 1.13 m, and reduced the positioning error to 0.051 m in the east direction.

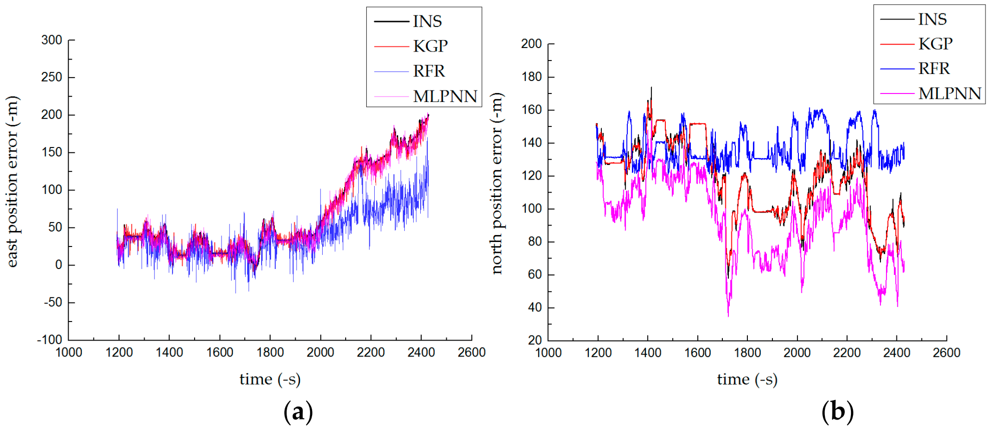


Figure 5. Prediction results of the normal driving test: (**a**) north position error and (**b**) east position error.

As shown in the figure, there was a significant cumulative error in the east direction, and the positioning error in the north direction was relatively stable. This is because the KGP method was set up with a positioning fix. Errors were compensated immediately when the parking time exceeded 5 s. For example, the positioning fix began working if the vehicle keeps moving forward at traffic light. The east direction corresponds to the y-axis of the inertial coordinate system, and the vehicle frequently turns and changes lanes after restarting at traffic lights. Therefore, the positional compensation effect of KGP in the east direction was not significant.

4.1.2. Results of the Parking Test

The positioning error shown in Figure 6 exhibited small fluctuations. Since the engine was still running, vehicle shaking could cause small errors in the INS. The prediction accuracies of the three methods were not much different, and the overall performance was better than that of INS alone. The prediction results of the GBDT and RFR were better than that of MLPNN because the initial weights and thresholds of the MLPNN were random, resulting in different results for each calculation.

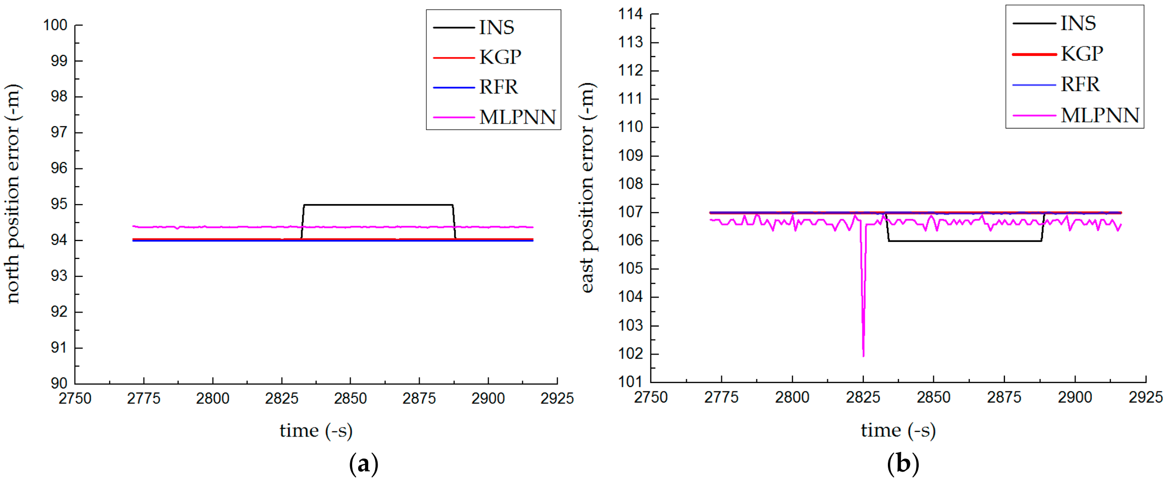


Figure 6. Prediction results of the parking test: (**a**) north position error and (**b**) east position error.

4.1.3. Results of the Combined Test

The combined test used the training model in the driving phase to predict INS errors in driving (motion) and parking (stationary) states. As shown in Figure 7, the RFR fluctuated greatly in the driving state, and the prediction accuracy was far lower than that in the parking state. When the vehicle's state in the training phase was inconsistent with that in the prediction phase, the generalization ability of the MLPNN was greatly weakened; the prediction was good only in the driving state. As a result, compared with other methods, the KGP algorithm had the highest positioning accuracy and could fit the trajectory curve well. The prediction was accurate and stable under various states. Table 4 shows the INS absolute error maxima in the driving and parking states in Test 3.

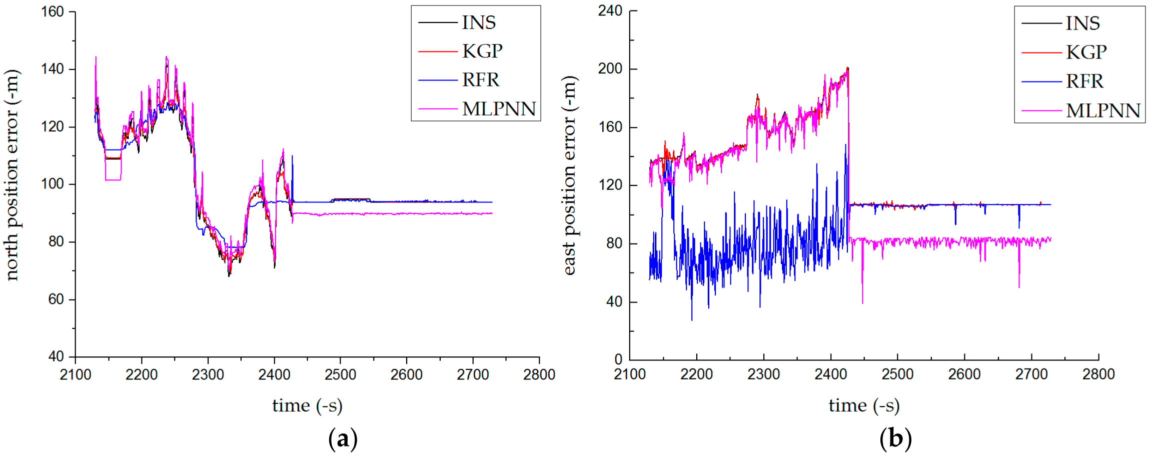


Figure 7. Prediction results of the combined test: (**a**) north position error and (**b**) east position error.

Table 4. Comparing errors of the combined test in east-north direction.

Position Error Value	**INS**		**KGP (m)**		**RFR (m)**		**MLPNN (m)**	
	North	**East**	**North**	**East**	**North**	**East**	**North**	**East**
Maximum (m)	142	201	138.66	201.67	128.64	151.43	144.52	198.82
Minimum (m)	68	107	68.97	106.50	78.22	27.55	70.53	81.06
Average (m)	103.78	156.41	103.78	155.45	103.93	79.46	105.51	152.35

Driving state.

Position Error Value	**INS**		**KGP (m)**		**RFR (m)**		**MLPNN (m)**	
	North	**East**	**North**	**East**	**North**	**East**	**North**	**East**
Maximum (m)	95	107	94.81	109.84	94.70	108.21	90.53	84.93
Minimum (m)	94	106	93.86	103.23	93.99	90.86	89.13	39.21
Average (m)	94.18	106.81	94.16	106.81	94.17	106.62	89.92	81.65

Parking State.

Compared with the predicted results of the normal driving test, with increased amounts of training data, the combined test could obtain more accurate positioning in a shorter time than the normal driving test.

4.2. Model Validity

The prediction models for the experiments described in Section 4.1 were applied to different roads (Figure 8). The system switched between training and prediction modes based on the actual conditions of the road, and the positioning prediction was divided into three phases. The blue area indicates good GPS navigation. The red area represents an occlusion test when the GPS signal was unstable and the accuracy was lowered due to high-rise buildings on both sides of the road. The blank area represents a tunnel test in which the GPS lost its lock due to the vehicle passing through a tunnel. Figure 9 shows the vehicle's velocity during 40 min of travel.

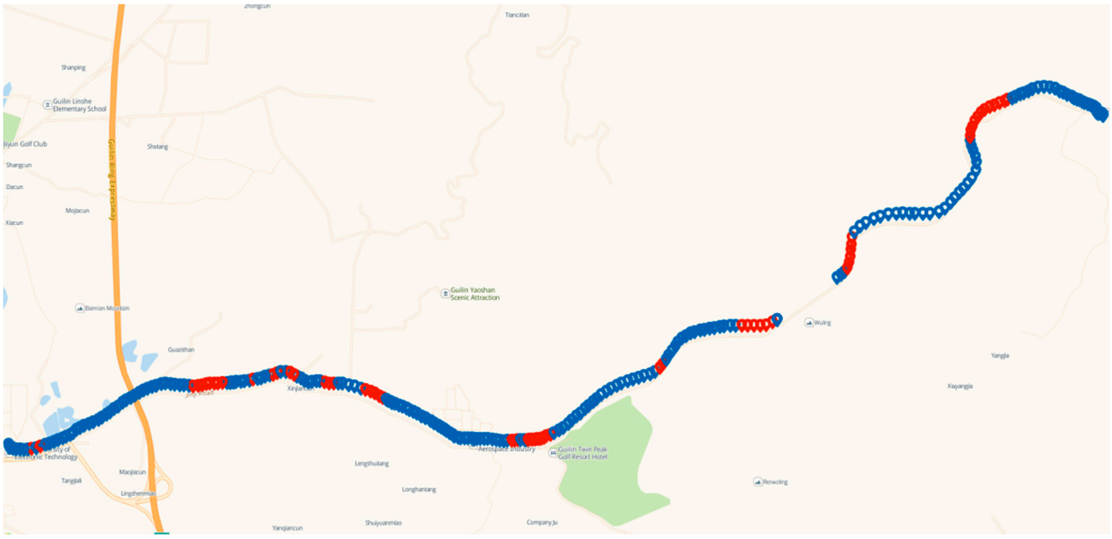


Figure 8. Experiment 2: Vehicle trajectory.

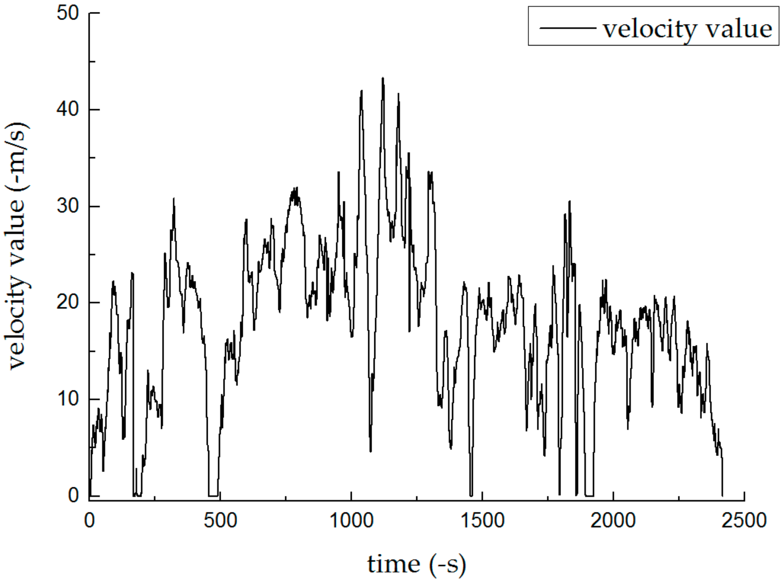


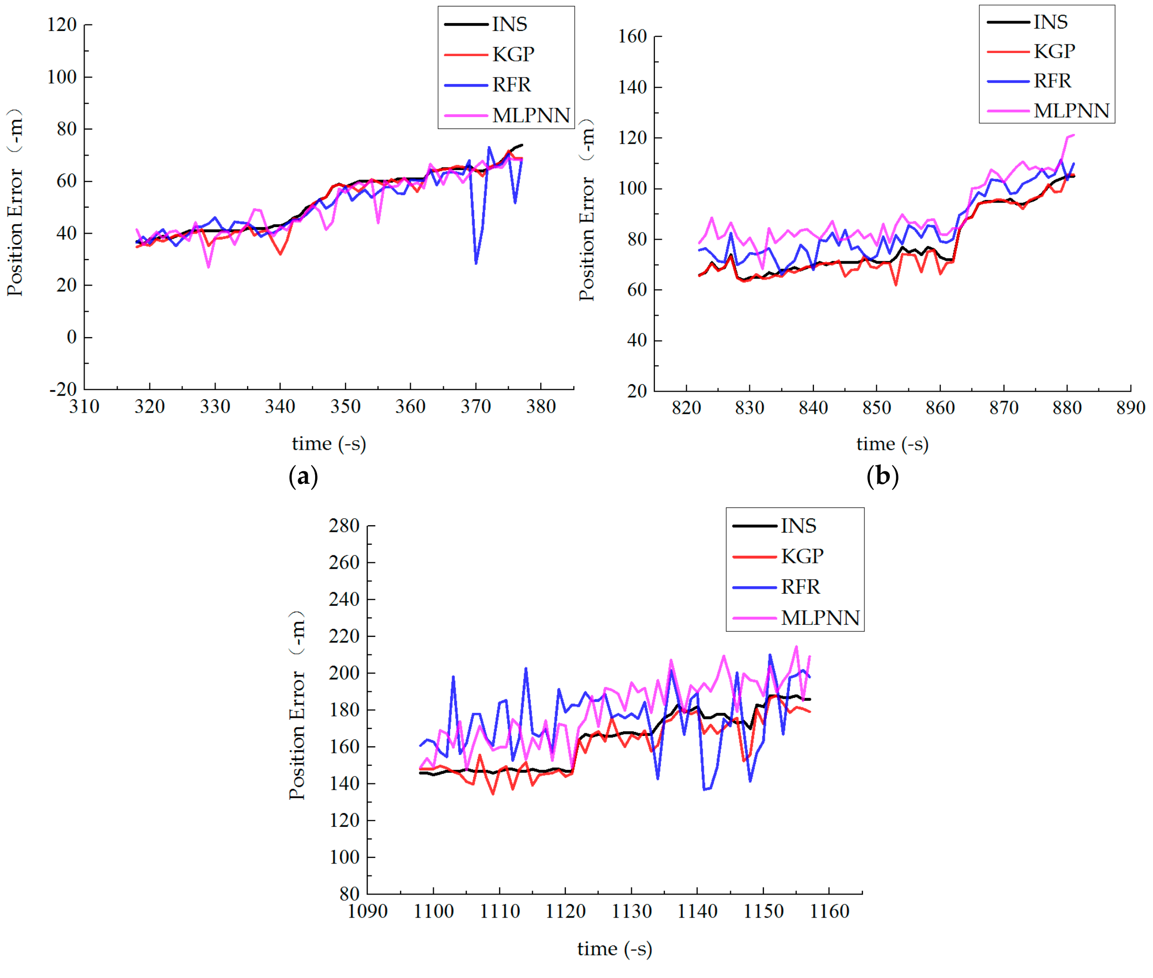
Figure 9. Experiment 2: Vehicle Speed.

Figure 10 depicts the final positioning error of these algorithms in three phases.

In phase 1, when the vehicle was driving on the city street which was straight and narrow for more than 5 min, the GPS signal was unstable. When the road was crowded, it was accompanied by frequent acceleration and deceleration, which enhanced the nonlinear relationship between input and output. Therefore, the prediction accuracy of MLPNN was low in the initial stage, RFR and KGP could fit the error curve well. But when the time was longer than 60 s, the prediction accuracy of RFR decreased, and KGP maintained strong generalization ability.

In phase 2, the vehicle entered the tunnel, and the INS error accumulation speed increased due to the complete loss of signals in the tunnel. It could be seen from Figure 10b that the prediction performance of KGP was higher than that of RFR and MLPNN algorithms. The maximum error was only 11.03 m, and the prediction result was stable.

In phase 3, the vehicle was driving on a curved road. The driving velocity and direction will change greatly with the curve trajectory, resulting in serious system cumulative error and less data set in a short time. Therefore, the prediction accuracy of the three methods was far less than that of phase 1 and phase 2, and there was a slight oscillation. But KGP still had obvious advantages compared with other methods, which was attributed to the improved KGP that could find out the relationship between the input and output, make up for the inherent error of the sensor, effectively model the position errors.



(**a**) (**b**)

(**c**)

Figure 10. Comparison results of different algorithms in position errors: (**a**) phase 1, (**b**) phase 2 and (**c**) phase 3.

To fully evaluate the effectiveness of the system's positioning data, we used the Root Mean Square Error (RMSE) to compare the predicted performance of the three methods. The RMSE is defined as follows (15):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{T} \left(\hat{y}_p - y_p\right)^2}{T}}, \tag{15}$$

where T is the GPS signal interruption time and $\hat{y}_p$ and y_p represent the predicted value and the actual output value, respectively.

Table 5 shows the RMSEs of different prediction algorithms. Compared with the RFR and MLPNN, the predicted result of the KGP algorithm had less error, and the predicted positioning data was closer to the reference data.

Table 5. Comparison results of RMSE with different algorithms.

Phase	KGP	RFR	MLPNN
Outage 1 (60 s)	2.63	6.80	4.63
Outage 2 (60 s)	5.02	7.00	11.88
Outage 3 (60 s)	6.28	22.64	16.03

Although the RFR and GBDT are both ensemble learning algorithms, the RFR continues to reduce the variance of the data set during the training phrases, while the GBDT improves the accuracy by reducing bias. Therefore, the RFR prediction produced larger errors, while the KGP prediction yielded better agreement with the data collected by the vehicle.

5. Conclusions

The KGP model can better meet the needs of accurate positioning during GPS signal interruption. The GBDT algorithm was integrated based on KF, and the regression parameters of the GBDT were optimized by the PSO algorithm to obtain a better prediction model. The road test showed that the generalization ability of the GBDT algorithm was stronger than that of a single algorithm in the integrated learning. The KGP model could effectively compensate for the cumulative error of the INS and correct the position data during GPS failures. The prediction remained accurate during a 20 min period. Compared with the existing methods, the positioning accuracy of our navigation solution was 28.20–59.89% higher than those of the MLPNN and RFR.

Author Contributions: H.Z. and T.L. proposed the idea and designed the experiments; T.L. and D.L. performed the experiments; T.L. wrote the paper; H.Z. and J.Z. guided the paper writing, reviewed the paper and offered the funding. L.Y. and Y.Z. revised the paper, discussed the idea and guided the paper writing. D.L. and F.P. collected and processed the data.

Funding: This work is supported by National Natural Science Foundation (61662013, U1501252, U1711263, 61662015, 61562014); Guangxi Innovation-Driven Development Project (Science and Technology Major Project)(AA17202024); The Guangxi Natural Science Foundation (2017GXNSFAA198372, 2016GXNSFAA380149); The Funds of Guangxi Key Lab of Trusted software Project(kx201511); The Teacher Growth Fund of the Education Development Foundation of Guangxi Normal University (EDF2015005); The Funds of Graduate student innovation program Guilin University of Electronic Technology (2017YJCX56, 2019YCXS045).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bijjahalli, S.; Gardi, A.; Sabatini, R. GNSS Performance Modelling for Positioning and Navigation in Urban Environments. In Proceedings of the 2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), Rome, Italy, 20–22 June 2018; pp. 521–526.
2. Huang, J.; Tan, H.S. A low-order DGPS-based vehicle positioning system under urban environment. *IEEE/ASME Trans. Mechatron.* **2006**, *11*, 567–575. [CrossRef]
3. Petritoli, E.; Leccese, F. Improvement of altitude precision in indoor and urban canyon navigation for small flying vehicles. In Proceedings of the 2015 IEEE Metrology for Aerospace (MetroAeroSpace), Benevento, Italy, 4–5 June 2015; pp. 56–60.
4. Li, X.; Xu, Q. A reliable fusion positioning strategy for land vehicles in GPS-denied environments based on low-cost sensors. *IEEE Trans. Ind. Electron.* **2017**, *64*, 3205–3215. [CrossRef]
5. Havyarimana, V.; Xiao, Z.; Wang, D. A novel probabilistic approach for vehicle position prediction in free, partial, and complete GPS outages. *Math. Prob. Eng.* **2015**, *2015*, 1–13. [CrossRef]
6. Li, X.; Chen, W.; Chan, C.; Li, B.; Song, X. Multi-sensor fusion methodology for enhanced land vehicle positioning. *Inf. Fusion* **2019**, *46*, 51–62. [CrossRef]
7. Trawny, N.; Mourikis, A.I.; Roumeliotis, S.I.; Johnson, A.E.; Montgomery, J.F. Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks. *J. Field Robot.* **2007**, *24*, 357–378. [CrossRef]
8. Angrisano, A.; Del Pizzo, S.; Gaglione, S.; Troisi, S.; Vultaggio, M. Using local redundancy to improve GNSS absolute positioning in harsh scenario. *ACTA IMEKO* **2018**, *7*, 16–23. [CrossRef]

9. Reimer, C.; Schneider, T.; Stock, M. INS/GNSS integration for aerobatic flight applications and aircraft motion surveying. *Sensors* **2017**, *17*, 941.
10. Bhatt, D.; Aggarwal, P.; Devabhaktuni, V.; Bhattacharya, P. A new source difference artificial neural network for enhanced positioning accuracy. *Meas. Sci. Technol.* **2012**, *23*, 105101. [CrossRef]
11. Aggarwal, P.; Bhatt, D.; Devabhaktuni, V.; Bhattacharya, P. Dempster Shafer neural network algorithm for land vehicle navigation application. *Inf. Sci.* **2013**, *253*, 26–33. [CrossRef]
12. Li, Z.K.; Wang, J.; Li, B.H.; Gao, J.X.; Tan, X.L. GPS/INS/Odometer integrated system using fuzzy neural network for land vehicle navigation applications. *J. Navigat.* **2014**, *67*, 967–983. [CrossRef]
13. Hasan, A.M.; Samsudin, K.; Ramli, A.R. Optimizing of ANFIS for estimating INS error during GPS outages. *J. Chin. Inst. Eng.* **2011**, *34*, 967–982. [CrossRef]
14. Qu, N.; Zhu, M.; Ren, Y.; Dou, S. Adaptive neuron-fuzzy inference system combined with principal components analysis for determination of compound thiamphenicol powder on near-infrared spectroscopy. *J. Taiwan Inst. Chem. Eng.* **2012**, *43*, 566–572. [CrossRef]
15. Jiang, C.; Chen, S.; Chen, Y.; Zhang, B.; Feng, Z.; Zhou, H.; Bo, Y. A MEMS IMU De-Noising method using long short term memory recurrent neural networks (LSTM-RNN). *Sensors* **2018**, *18*, 3470. [CrossRef] [PubMed]
16. Kurban, T.; Beşdok, E. A comparison of RBF neural network training algorithms for inertial sensor based terrain classification. *Sensors* **2009**, *9*, 6312–6329. [CrossRef]
17. Lei, X.; Li, J. An adaptive navigation method for a small unmanned aerial rotorcraft under complex environment. *Measurement* **2013**, *46*, 4166–4171. [CrossRef]
18. Li, J.; Song, N.; Yang, G.; Li, M.; Cai, Q. Improving positioning accuracy of vehicular navigation system during GPS outages utilizing ensemble learning algorithm. *Inf. Fusion* **2017**, *35*, 1–10. [CrossRef]
19. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
20. Han, S.; Wang, J. Integrated GPS/INS navigation system with dual-rate Kalman Filter. *GPS Solut.* **2012**, *16*, 389–404. [CrossRef]
21. Chen, X.Y.; Shen, C.; Zhang, W.B.; Tomizuka, M.; Xu, Y.; Chiu, K.L. Novel hybrid of strong tracking Kalman filter and wavelet neural network for GPS/INS during GPS outages. *Measurement* **2013**, *46*, 3847–3854. [CrossRef]
22. Eberhart, R.C.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS '95, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
23. Liu, H.; Tian, H.Q.; Li, Y.F.; Zhang, L. Comparison of four Adaboost algorithm based artificial neural networks in wind velocity predictions. *Energy Convers. Manag.* **2015**, *92*, 67–81. [CrossRef]
24. Zhang, Y.; Haghani, A. A gradient boosting method to improve travel time prediction. *Transp. Res. Part C* **2015**, *58*, 308–324. [CrossRef]
25. Li, L.; Yu, Y.; Bai, S.; Cheng, J.; Chen, X. Towards Effective Network Intrusion Detection: A Hybrid Model Integrating Gini Index and GBDT with PSO. *J. Sens.* **2018**, *2018*, 1578314. [CrossRef]
26. Chen, S.; Hong, X.; Harris, C.J. Particle Swarm Optimization aided orthogonal forward regression for unified data modeling. *IEEE Trans. Evol. Comput.* **2010**, *14*, 477–499. [CrossRef]
27. Samal, N.R.; Konar, A.; Das, S.; Abraham, A. A closed loop stability analysis and parameter selection of the particle swarm optimization dynamics for faster convergence. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 1769–1776.
28. Adusumilli, S.; Bhatt, D.; Wang, H.; Bhattacharya, P.; Devabhaktuni, V. A low-cost INS/GPS integration methodology based on random forest regression. *Expert Syst. Appl.* **2013**, *40*, 4653–4659. [CrossRef]
29. Adusumilli, S.; Bhatt, D.; Wang, H.; Bhattacharya, P.; Devabhaktuni, V. A novel hybrid approach utilizing principal component regression and random forest regression to bridge the period of GPS outages. *Neurocomputing* **2015**, *166*, 185–192. [CrossRef]