

Article

An Enhanced Map-Matching Algorithm for Real-Time Position Accuracy Improvement with a Low-Cost GPS Receiver

Jeong Min Kang¹, Han Sol Kim¹, Jin Bae Park^{1,*} and Yoon Ho Choi²

- ¹ Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Korea; kangjm@yonsei.ac.kr (J.M.K.); solsol@yonsei.ac.kr (H.S.K.)
- ² Department of Electronic Engineering, Kyonggi University, Suwon 16227, Kyonggi-Do, Korea; yhchoi@kyonggi.ac.kr
- * Correspondence: jbpark@yonsei.ac.kr; Tel.: +82-2-2123-2773

Received: 13 September 2018; Accepted: 5 November 2018; Published: 8 November 2018



Abstract: This paper proposes a real-time position accuracy improvement method for a low-cost global positioning system (GPS), which uses geographic data for forming a digital road database in the digital map information. We link the vehicle's location to the position on the digital map using the map-matching algorithm to improve the position accuracy. In the proposed method, we can distinguish the vehicle direction on the road and enhance the horizontal accuracy using the geographic data composed of the vector point set of the digital map. We use the iterative closest point (ICP) algorithm that calculates the rotation matrix and the translation vector to compensate for the disparity between the GPS and the digital map information. We also use the least squares method to correct the error caused by the rotation of the ICP algorithm and link on the digital map to eliminate the residual disparity. Finally, we implement the proposed method in real time with a low-cost embedded system and demonstrate the effectiveness of the proposed method through various experiments.

Keywords: low-cost GPS receiver; position estimation; ICP algorithm; map-matching; embedded system

1. Introduction

The development of the information technology(IT) industry has recently been applied to various fields, and has especially been influencing the development of the unmanned industry. The autonomous driving technology of unmanned vehicles is the core of this industry, and the development is accelerating existing automobile manufacturers. Various IT companies have also begun to pay attention to the smart car market. Unmanned ground vehicles for autonomous driving must judge and drive to the given destination. Four representative technologies are considered when performing a complete unmanned autonomous drive to a destination: the position estimation technology that can obtain the current position of the vehicle, the environment perception technology that can recognize the surrounding environmental information by utilizing various sensors, the obstacle avoiding technology that can move without collision to an object to the given destination, and the path planning technology that generates an optimal path to a destination in a given environment. The position estimation for finding the current position of a vehicle is particularly a basis for the autonomous driving system [1–4].

The majority of the contemporary position estimation technologies are generally based on the global navigation satellite system (GNSS), such as the global positioning system (GPS). Many studies and advances in technology have improved the accuracy of satellite-based location equipment.



A single-point position accuracy of GNSS is 10 m in open sky environment with many satellites and no degraded signal [5]. However, the performance of single GNSS cannot guarantee an accurate and continuous positioning in signal-degraded environments. In order to provide a remedy for the shortcomings of the single GNSS performance, a possible approach is the use of multi-constellation. The methods of the determination of the intersystem time offsets, which is characterized by a relative bias and drift between GPS and Galileo time scales, are proposed [6,7]. In particular, integrity monitoring has a great importance in safety critical operation, and the research for the performance assessment of the capabilities of integrity algorithms defined as receiver autonomous integrity monitoring (RAIM) are proposed [8,9]. In addition, the studies on improving the position accuracy, such as using the reference station that knows an accurate position, using additional sensors, and utilizing a digital map, have been actively performed.

An easy approach to compensate for these shortcomings is to use differential GPS (DGPS). DGPS is a method of inverting the errors contained in each satellite signal at the reference station, where the exact position is known, and improving the accuracy using a separate communication network. Carrier phase DGPS (CDGPS) is the augmentation system for carrier-phase measurements. This measurement obtains much less noise; however, its disadvantage is that the reliability problem occurs because of an integer ambiguity. A real-time kinematic-GPS method can be used for the spatial decorrelation in CDGPS, which improves the accuracy by efficiently calculating an integer ambiguity [10–12].

Other methods utilize additional equipment and sensors to perform the position estimation. Dead-reckoning using extra sensors, such as Inertial Measurement Unit (IMU), and speed measurements have been used to enhance the positioning precision and availability. In addition, an integrated navigation method combining GPS and IMU, a method by detecting lane using image information and lateral distance information of lane [13], matching the three-dimensional (3D) point cloud information generated during driving through Light Detection And Ranging(LIDAR) [14], and matching the road geometry information based on vision and radar sensors [15] have been proposed. However, these methods require separate communication links or additional infrastructure and sensors; hence, drawbacks of demanding a large amount of calculation and costs have been considered.

Another method for enhancing the position accuracy is the map-matching algorithm using a precise digital map. Most map-matching algorithms are tailored toward mapping the current position onto a vector representation of a road network to the closest node or the shape point, which is known as the point-to-point method [16]. The point-to-curve matching method also exists, which matches a location point to a curve closest to the network [17]. A weighting method of the link has been proposed to further improve the accuracy of the map-matching algorithm [18]. These methods are easy to perform, but the amount of computation and the position error increase as the link becomes more complex. Another geometric approach is to compare the trajectory of a vehicle with the road of the digital map. A method employing both distance and topology with low sampling rate, and considering the mutual influences of the GPS points in a trajectory, is proposed [19]. This method achieves better accuracy at a higher computational cost. A method for finding a moderate sampling rate utilizing user traveling behavior based on the observation that all possible segments on the map is proposed [20]. These methods have outperformed for single trajectory, but they are insufficient for real-time application with low sampling processes. However, a probabilistic map-matching method that can recover from mis-matches quickly [21], and an advanced map-matching algorithm that applies a fuzzy logic [22] is proposed. However, these algorithms still perform poorly when the sampling rate is low. A commonly used criterion for evaluating these trajectory map-matching methods is correct segment identification [23]. In order to determine the matching accuracy, the horizontal distance from the center of road is calculated. The horizontal errors in previous research are 9 m and 5.5 m [19,22]. However, these approaches cannot present the position error of each point. The methods of the map-matching algorithm based on additional sensors have also been presented. A localization method using a monocular camera and a 3D compact semantic map [24] and a map-based localization method

of requiring additional sensors and high computational complexity. Therefore, a method based on the iterative closest point (ICP) algorithm using only the GPS and digital map information has been proposed, which does not require additional sensors [26]. This method matches the disparity between the trajectory of the GPS and the digital map information, and is considered only in a single direction using the center of the road.

In this paper, we propose a method for enhancing the position accuracy of a low-cost GPS receiver based on map data without using any additional sensors. The points extracted from the roads prior to operating the algorithm are utilized as map data. They are stored in the form of vector data, including the position and direction information. We refer to these points as the vector points in the discussion that follows. A source point set is updated at each sampling time and contains the vector points of the vehicle trajectory acquired by the GPS. A reference point set is created from the vector points in the map point set, whose direction is parallel to those of the source point set at each sampling time. In this configuration, the ICP algorithm is employed to compute the transformation between the reference point set and the source point set. The vehicle trajectory is corrected after computing these errors. The position information is further corrected using the least squares method to compensate for the error caused by the rotation of the ICP algorithm and linked on the digital map to eliminate the residual disparity. Finally, we implement the proposed method in real time with a low-cost embedded system and demonstrate the effectiveness of the proposed method through various experiments.

2. Preliminaries

2.1. Specifications

The map-matching algorithm proposed herein can enhance the position accuracy. The characteristics of the proposed method are as follows:

- The performance of a low-cost GPS receiver is enhanced based on a GPS and the digital map information without using any additional sensors.
- The proposed method is simple. The necessary information includes the *x*-*y* coordinates, and heading. By ignoring the map data with no relation to the trajectory of a vehicle, computational efficiency can be enhanced. Thus, it is not limited by the computing power of a processor.
- The performance of the proposed method is verified through real-time experiments on a low-cost MCU-based embedded board.

2.2. Description of the Vector Point Sets

Under the assumption that a vehicle maneuvers along roads, the objective of the proposed method is to enhance the accuracy of low-cost GPS receivers by matching the acquired GPS trajectory with a pre-constructed road map database. These can be extracted by hand or automatically using image processing methods [27]. Once extracted, they are stored in the form of vector points, including the position and direction information. Herein, the *k*-th vector point \mathbf{v}_k consists of the position and the direction, which has the following form:

$$\mathbf{v}_k = (\mathbf{p}_k, \theta_k),\tag{1}$$

where $\mathbf{p}_k = (x, y)$; *x*, and *y* represent the latitude and longitude coordinates, respectively; and θ_k is the heading.

A vector point consists of the position and direction. The position is represented in the universal transverse mercator coordinate system. The heading is acquired by GPS information. In the case of a map database, it is obtained manually from the bearing on the map. The $\bar{\theta}_k$ has one of the following four values:

$$\overline{\theta}_{k} = \begin{vmatrix} East, & 45^{\circ} \leq \theta_{k} < 135^{\circ}, \\ South, & 135^{\circ} \leq \theta_{k} < 225^{\circ}, \\ West, & 225^{\circ} \leq \theta_{k} < 315^{\circ}, \\ North, & 315^{\circ} \leq \theta_{k} \text{ or } \theta_{k} < 45^{\circ}. \end{cases}$$

$$(2)$$

The proposed method utilizes the following vector point sets:

- 1. Assuming that a vehicle is traveling along the center of road, the map point set (MPS) is built prior to execution and composed of the vector points extracted from the center of roads.
- 2. The source point set (SPS) is updated from a GPS receiver at each sampling time and contains the vector points of the vehicle trajectory.
- 3. The reference point set (RPS) is created from the selected vector points in the MPS that a vehicle is expected to pass.

We present some examples of the MPS, SPS, and RPS as shown in Figure 1 to clarify the definitions of these vector point sets. In these examples, the SPS contains only the vector points, whose heading direction is east or south. Therefore, it is reasonable for the RPS to be constructed from the MPS vector points, whose heading direction is east or south. This example demonstrates why the RPS is introduced in the proposed algorithm.



Figure 1. Trajectories of a vehicle.

In short, the MPS is the pre-built fixed road map database, and the SPS and RPS are updated at each sampling time from the GPS receiver and the MPS, respectively. For simplicity, we will denote the MPS, SPS, and RPS as \mathcal{M} , \mathcal{S}_k , and \mathcal{R}_k , respectively, where k denotes the k-th sampling time. The MPS \mathcal{M} is further divided according to the direction, that is, $\mathcal{M} = \mathcal{M}^E \cup \mathcal{M}^S \cup \mathcal{M}^W \cup \mathcal{M}^N$, where each set only contains the vector points, whose heading direction is north, east, west, or south, respectively. Moreover, the *i*-th vector point v_i of any vector point set χ is represented as $v_i = (p_i, \overline{\theta}_i) = (\chi^P(i), \chi^H(i))$, where $\chi^P(i)$ and $\chi^H(i)$ are the position and heading of the *i*-th vector point of χ , respectively.

2.3. ICP-Based Map-Matching Method

Figure 2 shows the examples of the RPS, SPS, and MPS. The trajectories can be matched by rotating and translating one to fit the other. We can observe such trajectories in many scenarios. In this case, we can compute the rotational and translational errors between the trajectories using the ICP algorithm, which is widely used in field image processing. The ICP algorithm, which is a matching algorithm proposed by Besl and McKay [28], can be started with two point sets. Similarly, the GPS trajectory is composed of the vector points describing the trajectory of a vehicle for a certain period of time. The reference point set also consists of the vector points. In this configuration, the ICP algorithm is used for matching \mathbf{p}_k among these vector point sets. Given these sets, the ICP algorithm initiates a rotation matrix and a translation vector and finds a transformation matrix that contains a rotation matrix and a translation vector by minimizing the error function. The error function *J* is defined as follows:

$$E = \arg\min_{R,t} (\sum_{i=1}^{N_p} || m_i - Rp_i - t ||^2),$$
(3)

where m_i and p_i are the corresponding points, and N_p is the total number of points. The optimization problem can be solved by the iterative approach. A transformation matrix is obtained after the iteration procedure. We obtain the corrected GPS trajectory as shown in Figure 2 by correcting the rotational and translational errors. As can be seen, a low-cost GPS receiver provides a sufficiently reasonable performance when the ICP algorithm is employed. Thus, this study proposes a method to improve the accuracy of a low-cost GPS receiver based on the ICP algorithm.



Figure 2. Examples of the RPS, SPS, and MPS.

3. Enhanced Map-Matching Algorithm for Real-Time Position Accuracy Improvement

The main idea of the proposed method is to match the SPS S_k^P with the RPS \mathcal{R}_k^P , where $k \in \mathbb{Z}_{>0}$ denotes a sampling time. Figure 3 shows a flowchart of the proposed method, which could be divided into four main steps as follows:

- step 1. Loading of the digital databases,
- step 2. Calculation of the disparity information,
- step 3. Calibration of the rotational error using the least squares method,
- step 4. Location of the vehicle position on the map.



Figure 3. Flowchart of the proposed method.

3.1. Step 1: Loading of the Digital Databases

In the initialization stage, the MPS \mathcal{M} is loaded from the memory, and the variables required in the proposed method are declared and initialized. After initialization, the current position and heading of a vehicle are acquired from the GPS receiver through homegrown software and stored in the SPS \mathcal{S}_k at each sampling time k. The method skips to the GPS acquisition stage of the (k + 1)th sampling time before $N(\mathcal{S}_k) > buff$, where $N(\mathcal{S}_k)$ is the number of vector points stored in the SPS \mathcal{S}_k , and *buff* is the predefined scalar value. Figure 4 shows the examples of map-matching process. At k = 3, examples of \mathcal{M} and \mathcal{S}_3 are shown in Figure 4a to clarify the procedure.

The position correction method begins when $N(S_k) > buff$ is satisfied for a certain k. The first step is to create the RPS \mathcal{R}_k based on the MPS \mathcal{M} and the SPS $\mathcal{S}_k^H(i)$ for i = 1, 2, ..., buff. Algorithm 1 summarizes this process. We anticipate that a vehicle travels along a road, whose headings are parallel to that of a vehicle. Therefore, all of the MPS $\mathcal{M}(i)$, whose headings are equal to the SPS $\mathcal{S}_k^H(j)$ for all possible combinations of i and j, are chosen as the vector points of the RPS \mathcal{R}_k . Constructing the RPS \mathcal{R}_k is important for two reasons. First, the computational efficiency can be enhanced by ignoring the vector points of the MPS \mathcal{M} with no relation to the trajectory of a vehicle. Second, the horizontal accuracy is enhanced by considering the direction of a road. Figure 4b shows examples of \mathcal{R}_4 and \mathcal{S}_4 for buff = 4. In this example, \mathcal{S}_4 contains the vector points, whose heading directions are east and south. Thus, the method ignores the roads heading in directions to the west and north. \mathcal{R}_4 consists of $\mathcal{M}(i)$ whose $\mathcal{M}^H(i) = East$ or South (i.e., $\mathcal{R}_4 = \mathcal{M}^E \cup \mathcal{M}^S$).



(c)

Figure 4. Examples of the map-matching process: (a) \mathcal{M} and \mathcal{S}_k at k = 3; (b) \mathcal{R}_k and \mathcal{S}_k at k = 4; and (c) \mathcal{R}_k and $\overline{\mathcal{S}}_k$ at k = 4.

```
      Algorithm 1 Algorithm for constructing the RPS

      function Construction (S_k^H)

      Initialize: \mathcal{R}_k \leftarrow empty set

      for i \leftarrow 1 to buff do

      if S_k^H(i) is North then

      \mathcal{R}_k = \mathcal{R}_k \cup \mathcal{M}^N

      \mathcal{R}_k = \mathcal{R}_k \cup \mathcal{M}^S

      else if S_k^H(i) is South then

      \mathcal{R}_k = \mathcal{R}_k \cup \mathcal{M}^S

      \mathcal{R}_k = \mathcal{R}_k \cup \mathcal{M}^S

      end if

      if \mathcal{S}_k^H(i) is East then

      \mathcal{R}_k = \mathcal{R}_k \cup \mathcal{M}^E

      end if

      end if

      end if

      end for

      Return \mathcal{R}_k

      end function
```

3.2. Step 2: Calculation of the Disparity Information

After constructing \mathcal{R}_k , the ICP algorithm is employed to compute the rotational and translational errors between \mathcal{R}_k^P and \mathcal{S}_k^P . The ICP algorithm outputs the rotation matrix and the translation vector in the following forms:

$$\mathbf{R} = \begin{pmatrix} \cos\theta_E & -\sin\theta_E \\ \sin\theta_E & \cos\theta_E \end{pmatrix}, \mathbf{T} = \begin{pmatrix} x_E \\ y_E \end{pmatrix},$$
(4)

where θ_E is the rotational error between two vector point sets, and x_E and y_E are the translational errors of each axis.

The estimation value of reference can be derived by calculating the transformation to the S_k^p . Using Label (4), S_k is corrected as follows:

$$\overline{\mathcal{S}}_{k}^{P}(i) = \mathbf{R}\mathcal{S}_{k}^{P}(i) + \mathbf{T}, \text{ for } i = 1, 2, ..., \text{ buff},$$
(5)

where $\overline{\mathcal{S}}_{k}^{P}(i)$ is the corrected trajectory of a vehicle.

Figure 4c shows \mathcal{R}_4 and $\overline{\mathcal{S}}_4$, from which we can conclude that the horizontal position error is minimized.

3.3. Step 3: Calibration of Rotational Error using the Least Squares Method

In the proposed method, the ICP is performed by maintaining a constant number of buffer and shifting the information from which the new GPS information is obtained. The ICP algorithm returns the one transformation between \mathcal{R}_k^P and \mathcal{S}_k^P . This transformation does not sufficiently reflect the change value between the trajectory and the GPS information, which are newly obtained. The ICP algorithm is more sensitive to rotation than translation [29]; hence, we use the least squares method to calibrate the rotational error. From $\overline{\mathcal{S}}_k^P$, which is the corrected trajectory of a vehicle, the output equation is as follows [30]:

$$z_k = H_k x_k + w_k, \tag{6}$$

where z_k is the latest point of \overline{S}_k^P , H_k is the output matrix, x_k is the state vector, and w_k is the measurement noise. We can then compute the state estimation \hat{x}_k that minimizes the cost function J, where J is given as follows:

$$J = (z_k - H_k \hat{x}_k)^T (z_k - H_k \hat{x}_k) = z_k^T z_k - \hat{x}_k^T H_k^T z_k - z_k^T H_k \hat{x}_k + \hat{x}_k^T H_k^T H_k \hat{x}_k.$$
 (7)

We compute the partial derivative as equal to zero to minimize J with respect to \hat{x}_k :

$$\frac{\partial J}{\partial \hat{x}_k} = -z_k^T H_k - z_k^T H_k + 2\hat{x}_k^T H_k^T H_k = 0.$$
(8)

Solving (8) for \hat{x}_k , the state estimation is as follows:

$$\hat{x}_k = (H_k^T H_k)^{-1} H_k^T z_k.$$
(9)

In this procedure, the state estimation \hat{x}_k implies the current position of a vehicle.

3.4. Step 4: Location of the Vehicle Position on the Map

The state estimation \hat{x} is linked to \mathcal{R}_k to eliminate the residual disparity. The projection method uses the vector projection theorem, which is from the current state estimation and the two points near the estimation point. The lateral error is reduced, and the vehicle position is located on the road of the digital map, implying that the accuracy of the low-cost GPS receiver has been enhanced.

Finally, the FIFO buffer, S_k , is shifted, and the method waits until a new GPS point is received. The memory requirement does not increase as $N(S_k)$ is confined to *buff*, even if the proposed method operates for a long time. The execution time is mainly affected by $N(\mathcal{R}_k)$ and $N(\mathcal{S}_k)$. $N(\mathcal{S}_k)$ plays an important role in the proposed method; hence, larger values of $N(\mathcal{S}_k)$ not only enhance the performance of the method, but also increase the computational load. Thus, this parameter should be chosen according to the desired performance of the proposed method. In addition, the computational burden increases with the map size, which can be solved by confining the region of \mathcal{R}_k based on the current position of a vehicle.

4. Experimental Setup and Results

4.1. Hardware Implementation

The proposed method is computationally efficient; hence, it can be implemented using a low-cost embedded system. We developed the embedded system as shown in Figure 5, in order to demonstrate the efficiency of the proposed method. This system consisted of a low-cost processor, a secure digital (SD) card slot, and a Bluetooth module. The main processor was an STM32F405VGT6 (ST Microelectronics, Inc., Geneva, Switzerland) [31], which operates at a frequency of 168 MHz, and has a static RAM of 192 Kbytes. The SD card was used to store the experimental results and \mathcal{M} . The Bluetooth module was employed to receive GPS points from a smartphone. A Galaxy A5 (Samsung Electronics, Inc., Suwon, Korea) smartphone, which provides a trajectory with a horizontal accuracy of up to 10 m, was connected to our system through the Bluetooth connectivity. The sampling time of the proposed method is set to 1 Hz, that is, one GPS point per second is sent to the embedded board.



Figure 5. System block diagram and photograph of the developed embedded system.

4.2. Experimental Setup

We employed a high-end GPS/INS device (NovAtel, Inc., Calgary, Canada) composed of UIMU-H58 [32], SPAN-SE-D-GENERIC [33], and GPS-703-GGG [34] to obtain the actual trajectory of a vehicle. In this configuration, the device provides the position of a vehicle with an accuracy of 0.4 m; thus, it is reasonable to assume that the position acquired by the device is the ground truth.

The main purpose of our experiments was to verify that a trajectory acquired by a low-cost GPS receiver is effectively corrected by the proposed method. We also showed that implementing the proposed method in real time was possible using the low-cost embedded system. The GPS/INS device, a smartphone, and an embedded board were positioned on the vehicle as shown in Figure 6.

We performed experiments for the six courses shown in Figure 7. Table 1 lists the length of the paths and the number of the GPS points for each experiment.





Figure 6. Setting of the experimental devices: (a) an antenna on the vehicle and (b) IMU in the vehicles.



Figure 7. Overall map comprising six courses.

Table 1. Length and number of GPS points of the experimental courses.

	Course 1	Course 2	Course 3	Course 4	Course 5	Course 6
Length (km)	1.46	1.82	1.85	5.55	2.15	1.67
Number of GPS points	220	249	212	647	281	191

4.3. Experimental Results

As mentioned in Section 4.2, we assumed that the position acquired by the high-end GPS/INS device was the ground truth, and calculated the position errors of the low-cost GPS and the proposed method, respectively. Figures 8–13 show the six courses on real routes containing curved roads and intersections, including the low-cost GPS measurements, the final corrected trajectories through the proposed method, and the ground truth. Each figure contained the overall trajectories and four magnified subsections.

Figure 8. Experimental results for Course 1.

Figure 9. Experimental results for Course 2.

Figure 10. Experimental results for Course 3.

Figure 11. Experimental results for Course 4.

Figure 12. Experimental results for Course 5.

Figure 13. Experimental results for Course 6.

These figures show that, for all cases, including straight lanes, curved lanes, and intersections, the corrected trajectories were located much closer to the ground truth compared to those originally given by the low-cost GPS receiver. Figures 8c and 11b,c also depict that the trajectories acquired by the low-cost GPS receiver were located on the opposite side of the roads; however, the proposed

method determined which side of the roads a vehicle was traveling on. If the vehicle traveled along the center of road, the trajectory from the high-end GPS/INS device would be on the RPS. Therefore, the horizontal error between the points acquired by the proposed method and the high-end GPS/INS device was close to zero. However, the corrected trajectory sometimes cannot follow the curved lanes as the accumulated GPS points are used to correct the trajectory (e.g., Figure 13c).

The error distributions of the proposed method and the raw GPS points for the six courses were summarized in Figure 14 to quantitatively evaluate the performance of the proposed method. In this figure, the horizontal axis represents the error that is calculated in meters. The blue graph denotes the error distribution that explains how many times each error occurred, while the red line shows the mean of each error distribution. The proposed method generated much smaller distance error than the raw GPS. The mean and the standard deviation of these error distributions were listed in Table 2 for further clarification. The columns labeled 'Avg. (GPS)' and 'Avg. (Prop.)' represent the root mean square error between the position acquired by the GPS/INS device and that of the low-cost GPS receiver and the proposed method, respectively. The 'Std.' columns list the standard deviations of each error distribution. The errors in the 'Avg. (Prop.)' column are mainly from the vertical errors because the proposed method only improved the horizontal accuracy. However, the merit of the proposed method remains in that it reduces the horizontal error. The 'Improv.' represents the improvement rate which is obtained by dividing the difference between 'Avg. (GPS)' and 'Avg. (Prop.)' by 'Avg. (GPS)', from which we can conclude that the proposed method improves the accuracy of the low-cost GPS receiver by up to 62.2927%.

We ignored the map data with no relation to the trajectory of a vehicle by utilizing the RPS, so the map size used for map-matching was reduced. The data from the low-cost GPS and high-end GPS/INS device was updated once per second, and all data was transferred to the laptop. The proposed method was executed on the laptop in real time and updated the position without any time delay. The average running time for one single point according to the RPS were listed in Table 3. The row labeled 'w/o RPS' and 'w/ RPS' represent the average process time of the proposed method for each point, taking into account the RPS or not, respectively. It took approximately 0.066 s on a single point without utilizing the RPS, while 0.043 s with the RPS for Course 2. As the map size decreased, a faster average running time of the proposed method was achieved.

We mentioned that the performance of the proposed method is mainly dependent on the size of *buff*. Varying *buff* from 5 to 20 in Step 5, we measured the average and the standard deviation of the root mean square errors for Course 2. The results in Table 4 indicated that the proposed method provided a better performance as the size of *buff* increased, and the graphical representation of this result is shown in Figure 15. However, the increased size of *buff* increased the computational burden of operating the proposed method; thus, *buff* should be chosen according to the required accuracy and processor performance.

Course	Avg. (GPS) (m)	Std. (GPS) (m)	Avg. (Prop.) (m)	Std. (Prop.) (m)	Improv. (%)
Course 1	3.6013	1.7243	2.0023	1.1261	44.4006
Course 2	4.1199	1.3088	1.5535	0.9537	62.2927
Course 3	3.7037	1.1829	1.9854	1.0841	46.3941
Course 4	3.7408	2.1175	2.0153	2.2406	46.1264
Course 5	2.2805	0.8799	1.5031	0.6587	34.0890
Course 6	3.6874	1.5508	1.4432	1.2074	60.8613

Table 2. Comparison results.

 Table 3. The average running time for a one single point according to the RPS.

	Course 1	Course 2	Course 3	Course 4	Course 5	Course 6
w/o RPS (s)	0.059	0.066	0.056	0.178	0.071	0.049
w/ RPS (s)	0.036	0.043	0.034	0.119	0.046	0.031

Figure 14. Error histograms. The blue bars represent the error distributions, while the red line marks the average value of each error distribution.

Table 4. Mean and standard deviations of Course 2 according to the size of *buff*.

Figure 15. Mean and standard deviations according to the size of *buff*.

5. Discussion

The proposed algorithm assumes that a vehicle is traveling along the roads, which enables us to minimize the horizontal error, but does not correct the vertical error, as can be seen from the results. Minimizing the horizontal error is important because one can clearly determine which side of the road a vehicle is actually traveling on. Thus, the proposed method provides an advantage in the fields of car navigation and autonomous vehicle guidance.

A number of methods providing sub-meter accuracy have recently been studied. However, these methods require additional sensors or complicated algorithms, which increase the implementation cost. Since the proposed method uses only a low-cost GPS receiver and map data, the computational efficiency can be enhanced in real time. In addition, more accurate trajectories are anticipated to be obtained by applying additional sensor fusion algorithms to the trajectory corrected by the proposed method.

6. Conclusions

In this paper, we proposed a method for improving the position accuracy of a low-cost GPS receiver for car navigation systems using pre-built map data. In our method, the position and the direction of the points extracted from the roads were utilized as the map data. The ICP method was employed to match the points between the roads and the vehicle trajectory. Moreover, only those roads parallel to the vehicle trajectory were used rather than the entire roads. We also used the least squares method to correct the error caused by the rotation of the ICP algorithm and linked on the digital map to eliminate the residual disparity. Finally, we performed the proposed method in real time with a low-cost embedded system. Consequently, we verified that the proposed method improved the position accuracy of the low-cost GPS in real time.

Author Contributions: J.M.K. conceived and designed the experiments; H.S.K. and J.M.K. performed the experiments; J.B.P. and Y.H.C. analyzed the data; and J.M.K., H.S.K., J.B.P. and Y.H.C. wrote the paper.

Funding: This research received no external funding.

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

References

- 1. Shin, J.; Huh, J.; Park, Y. Asymptotically stable path following for lateral motion of unmanned ground vehicle. *Control Eng. Pract.* **2013**, *40*, 102–112. [CrossRef]
- 2. Thrun, S. Toward robotic cars. Commun. ACM 2010, 53, 99–106. [CrossRef]
- 3. Kim, G.; Lee, D.G.; Kim, B. Path planning for avoiding obstacles for unmanned ground vehicles. In *Advanced Multimedia and Ubiquitous Engineering*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 354, pp. 211–217.
- 4. Kim, D.; Oh, S.M.; Rehg, J.M. Traversability classification for UGV navigation: A comparison of patch and superpixel representations. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 3166–3173.
- 5. Kaplan, E.; Hegarty, C. Understanding GPS: Principles and Applications; Artech House: Norwood, MA, USA, 2005.
- 6. Gioia, C.; Borio, D. A statistical characterization of the Galileo-to-GPS inter-system bias. *J. Geod.* **2016**, *90*, 1279–1291. [CrossRef]
- 7. Gioia, C. GNSS Navigation in Difficult Environments: Hybridization and Reliability. Ph.D. Thesis, University Parthenope of Naples, Naples, Italy, 2014.
- 8. Angrisano, A.; Gaglione, S.; Gioia, C. Perfermance assessment of aided Global Navigation Satellite System for land navigation. *IET Radar Sonar Navig.* **2013**, *7*, 671–680. [CrossRef]
- 9. Angrisano, A.; Gaglione, S.; Del Core, G.; Gioia, C. GNSS reliability testing in signal-degraded scenario. *Int. J. Navig. Obs.* **2013**, 2013, 870365. [CrossRef]
- 10. Rezaei, S.; Sengupta, R. Kalman filter-based integration of DGPS and vehicle sensors for localization. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 1080–1088. [CrossRef]
- 11. Busse, F.D.; How, J.P.; Simpson, J. Demonstration of adaptive extended kalman filter for low-earth-orbit formation estimation Using CDGPS. *Navigation* **2003**, *50*, 19–93. [CrossRef]
- 12. Tamura, Y.; Matsui, M.; Pagnini, L.C.; Ishibashi, R.; Yoshida, A. Measurement of wind-induced response of building using RTK-GPS. *J. Wind Eng. Ind. Aerod.* **2012**, *49*, 18–22. [CrossRef]
- 13. Lee, B.H.; Im, S.H.; Heo, M.B.; Jee, G.I. Curve-Modeled lane detection based GPS lateral error correction enhancement. *J. Inst. Control Robot. Syst.* **2015**, *21*, 81–86. [CrossRef]
- Levinson, J.; Thrun, S. Robust vehicle localization in urban environments using probabilistic maps. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 4372–4378.
- 15. Serfling, M.; Schweiger, R.; Ritter, W. Road course estimation in a night vision application using a digital map, a camera sensor and a prototypical imaging radar system. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 810–815.
- 16. Bernstein, D.; Kornhauser, A. *An Introduction to Map Matching for Personal Navigation Assistants*; The National Academies of Sciences, Engineering, and Medicine: Washington, WA, USA, 1998.
- 17. White, C.E.; Bernstein, D.; Kornhauser, A.L. Some map matching algorithms for personal navigation assistants. *Transp. Res. Part C Emerg. Technol.* **2000**, *8*, 91–108. [CrossRef]
- 18. Jimenez, F.; Monzon, S.; Naranjo, J.E. Definition of an enhanced map-matching algorithm for urban environments with poor GNSS signal quality. *Sensors* **2016**, *16*, 193. [CrossRef] [PubMed]
- 19. Lou, Y.; Zhang, C.; Zheng, Y.; Xie, X.; Wang, W.; Huang, Y. Map-matching for low-sampling-rate GPS trajectories. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information System, Seattle, WA, USA, 4–6 November 2009; pp. 352–361.
- 20. Li, Y.; Huang, Q.; Kerber, M.; Zhang, L.; Guibas, L. Large-scale joint map matching of GPS traces. In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Orlando, FL, USA, 5–8 November 2013; pp. 214–223.
- 21. Ochieng, W.Y.; Quddus, M.A.; Noland, R.B. Map-matching in complex urban road networks. *Braz. J. Cartogr.* **2003**, *55*, 1–14.
- 22. Quddus, M.A.; Noland, R.B.; Ochieng, W.Y. A high accuracy fuzzy logic-based map-matching algorithm for road transport. *J. Intell. Transp. Syst.* **2006**, *10*, 103–115. [CrossRef]
- 23. Hashemi, M.; Karimi, H.A. A critical review of real-time map-matching algorithms: Current issues and future directions. *Comput. Environ. Urban Syst.* **2014**, *48*, 153–165. [CrossRef]
- 24. Xiao, Z.; Jiang, K.; Xie, S.; Wen, T.; Yu, C.; Yang, D. Monocular vehicle self-localization method based on compact semantic map. *arXiv* **2018**, arXiv:1805.06155.

- 25. Wang, L.; Zhang, Y.; Wang, J. Map-based localization method for autonomous vehicles using 3D-LIDAR. *IFAC-PapersOnLine* **2017**, *50*, 276–281. [CrossRef]
- 26. Choe, T.S.; Kang, J.M.; Kim, H.S.; Park, J.B. Low-end GPS position accuracy enhancement method by using map information. *Trans. Korean Inst. Electr. Eng.* **2017**, *65*, 659–665. [CrossRef]
- 27. Chiang, Y.Y.; Knoblock, C.A. Automatic extraction of road intersection position, connectivity, and orientations from raster maps. In Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Irvine, CA, USA, 5–7 November 2008; pp. 1–10.
- Besl, P.J.; Mckay, N.D. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*; International Society for Optics and Photonics: Bellingham, WA, USA; 1992; Volume 1611, pp. 586–607.
- 29. Specht, A.R.; Devy, M.; Sappa, A. Robustness to Initial Conditions in Registration of Range Images Using the ICP Algorithm. Available online: http://www.cvc.uab.es/~asappa/publications/C_ISRA.pdf (accessed on 8 November 2018).
- 30. Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches;* John Wiley & Sons: Hoboken, NJ, USA, 2006.
- 31. STMicroelectronics Inc. STM32F405VG. Available online: http://www.webcitation.org/6hjayc5XG (accessed on 7 November 2018).
- 32. NovAtel Inc. UIMU-H58. Available online: http://www.webcitation.org/6hjaRYLku (accessed on 7 November 2018).
- 33. NovAtel Inc. SPAN-SE. Available online: http://www.webcitation.org/6hjarC9o9 (accessed on 7 November 2018).
- 34. NovAtel Inc. GPS-703-GGG. Available online: http://www.webcitation.org/6hjcvSknC (accessed on 7 November 2018).

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).