Vehicle localization by dynamic programming from altitude and yaw rate time series acquired by MEMS sensor

Takayoshi Yokota

Department of Information and Electronics, Graduate School of Engineering, Tottori University, Tottori, Japan

ABSTRACT

Localization technology is mostly achieved through Global Navigation Satellite Systems (GNSS). However, in mountainous areas where leaves and trees block signals from satellites, getting an appropriate location with GNSS positioning sometimes becomes difficult. In this paper, a localization algorithm based on altitude profiles with angular momentum (yaw rate) profile is proposed. It is a kind of landmark-based localization algorithm in which altitude and angular momentum (yaw rate) profiles are treated as landmarks. The proposed localization algorithm attempts to identify the current position by matching altitude and angular momentum (yaw rate) reference data prepared in advance with altitude and angular momentum (yaw rate) data currently acquired. The effectiveness of the algorithm is presented with several experimental results obtained with real data. This paper is an extension to the previous paper presented at the SICE Annual Conference 2020 [Yokota. Localization algorithm based on altitude time series in GNSS-denied environment. In: 2020 Proceedings of the SICE Annual Conference; Chiang Mai, Thailand; 2020 Sep 23–26. p. 952–957], in which vehicle localization was carried out by using only altitude profile matching. This paper used both altitude and angular momentum (yaw rate) for sensor fusion to improve the accuracy of the localization.

1. Introduction

With the spread of smart phones and car navigation systems, various applications (including various outdoor sports) using positioning results obtained with Global Navigation Satellite Systems (GNSS) are spreading. Depending on the quality of the reception of radio waves, users may experience an increase in positioning errors. In this study, the author proposes an algorithm for measuring the location of vehicles where GNSS positioning is difficult [1]. The position data is useful for assessing traffic or analysing road usage [2,3]. In recent years, it has become possible to easily use altitude and angular momentum (yaw rate) data. Specifically, altitude is estimated from pressure information obtained from pressure sensors. Angular momentum (yaw rate) data can be obtained by gyroscopes. In recent years, with the spread of acceleration sensors, there have been several research reports on estimating the position of a vehicle by capturing the characteristics of road surface properties when a vehicle is traveling. With these algorithms, the position of a vehicle is estimated by acquiring the properties of the road surface in real time with an acceleration sensor and calibrating it with vibration data [4,5] or pitch data [6–9] and odometer data collected in advance. As a similar approach, a road profile is estimated by inertial measurement unit (IMU) data, and its spatial spectrum for each road segment (spectrogram plot) is used as a feature [10,11].

However, there are likely to be many issues such as the dependence on vehicles and the need to respond to changes in the road surface over time. Therefore, a new localization algorithm is proposed. It is a position estimation algorithm that attempts to identify a current position by matching altitude and angular momentum (yaw rate) reference data prepared in advance with corresponding data currently acquired.

Landmark-based localization is a new technique that deals with suitable objects in a scene of the road that can be used as features for identifying the location of a vehicle or user. Image processing-based localization including 3D LiDAR depth maps can be effective [12,13]; however, the cost of preparing data is not small. In this paper, altitude and angular momentum, and especially yaw rate, data obtained while driving on roads are used as features. This paper is an extension to a previous SICE Annual Conference paper [1] in that a new experiment is conducted with finer ground truth data by using a real-time kinematic (RTK)-GNSS and sensor fusion with angular momentum (yaw rate) data that is obtained by a micro electro mechanical systems (MEMS) sensor to improve the accuracy of localization, especially for regions with little variation in altitude. The author also added more detailed descriptions of the
algorithm. The original method [1] was based only on altitude data. This time, it is extended to incorporate yaw rate data as additional road features and unwanted-stop estimation are also incorporated to improve the location estimation accuracy. The method for optimum estimation is based on dynamic programming, which was also used in the previous paper [1].

2. Estimating location with atmospheric pressure sensor and yaw rate sensor

In this section, location estimation based on altitude and yaw rate is proposed. The altitude is estimated from atmospheric pressure sensor and yaw rate is measured by a MEMS gyroscope in a smartphone. Both types of data are used as features of the road, and the location estimation is based on these features.

2.1. Estimating altitude time series by atmospheric pressure sensor

It is known that altitude can be estimated from atmospheric pressure by using the following equation. Formulations can be found in some literature [14]; however, the following reformulation is provided in order to help avoid confusion.

\[ dp = -\rho g \, dz, \]  

(1)

where \( dp \) is the differential of atmospheric pressure, \( g \) is the gravitational constant of Earth, \( \rho \) is the atmospheric density, and \( dz \) is the differential of altitude. On the other hand,

\[Mp = \rho RT, \]

(2)

where \( M \) is a mole of atmosphere and \( R \) is the gas constant. Substituting this into Equation (1), we get the following.

\[ RT \, dp = -Mpg \, dz \]

(3)

Here, the temperature \( T \) depends linearly on the altitude \( z \). That is, as follows.

\[ T(z) = T(z_0) - \gamma z \]

(4)

\[ \frac{1}{\rho} \, dp = -\frac{Mg}{R} \frac{1}{(T(z_0) - \gamma z)} \, dz \]

(5)

Solving the differential equation, we obtain the following.

\[ h = \left( \left( \frac{p_0}{\rho} \right)^{\frac{\gamma}{R}} - 1 \right) T(z_0) \]

(6)

With this equation, the altitude of any point can be estimated from the atmospheric pressure. A MEMS pressure sensor is embedded in most smartphones. The author assumes that reference altitude data is available in the form of altitude probe data obtained by a vehicle. In this case, the altitude data that is obtained from the atmospheric pressure data taken by a vehicle is time series data. The velocity of a vehicle does not have to be estimated during the estimation of the location. Rather, the time difference between the current time series and the reference time series when the reference vehicle and the vehicle of interest are at the same position is estimated.

2.2. Yaw rate time series data obtained by a gyroscope

As another information source, the author thought that angular momentum, especially yaw rate, could be a candidate for a feature of roads because left and right turn manoeuvre and driving along curves will be reflected in the yaw rate efficiently compared with pitch and roll. MEMS gyroscopes are available and are embedded in most smartphones. Angular momentum data such as the yaw rate can be measured at more than 50 Hz by these MEMS gyroscopes.

2.3. Similarity matching of altitude profiles in time domain

First, a localization algorithm that uses similarity matching of altitude profile [1] is described in this section. An extension to a sensor fusion of altitude and angular momentum (yaw rate) is described in the next subsection.

Altitude data time series are compared with the previously obtained reference time series in the time domain. The reference altitude time series are sampled at discrete time \( t_{ref}(i) \) and the current altitude time series are sampled at discrete time \( t_{cur}(i) \). Because, the position of both types of altitude data should be the same, \( t_{ref}(i) \) and \( t_{cur}(i) \) must be associated with the time difference \( d(i) \) that is caused by the difference in the velocity and traffic conditions of the reference and current vehicle.

\[ t_{ref}(i) = i \Delta t + d(i), \]

(7)

\[ t_{cur}(i) = i \Delta t, \]

(8)

\[ d(i) = t_{ref}(i) - t_{cur}(i), \]

(9)

\[ i = 0, 1, \ldots, P - 1, \]

(10)

where \( t_{ref}(i) \) and \( t_{cur}(i) \) are discrete sampling times of the altitude data for the reference and current altitude data. Both sampling times are measured from the origin of their trips. \( d(i) \) is the time difference between the sampling times of the reference vehicle and current vehicle. \( P \) is the number of pieces of altitude time series data.

As described in Figure 1, the location of the current vehicle, vehicle B, can be estimated by similarity
matching to the reference altitude profile. The measure function $E(t_{cur})$ to be minimized is defined with a continuous-time system formulation as follows.

$$E(t_{cur}) = \int_{0}^{t_{cur}} \left| f(t_{ref}(\tau)) - g(\tau) \right|^2 d\tau$$  \hspace{1cm} (11)

$$t_{ref}(t_{cur}) = t_{cur} + d(t_{cur})$$  \hspace{1cm} (12)

$$d(t_{cur}) = t_{ref}(t_{cur}) - t_{ref}(t_{cur})$$  \hspace{1cm} (13)

where, $t_{ref}(t_{cur})$ means that the position of the vehicle for the 1st run at $t_{ref}(t_{cur})$ is the same as that of the vehicle for the 2nd run at $t_{cur}$. Equation (11) measures the accumulated squared error between the two altitude profiles from the origin up to time $t_{cur}$. The goal of the problem is to find an optimum time difference profile $d(\tau)$, $0 \leq \tau \leq t_{cur}$ so that the measure $E(t_{cur})$ is minimized for all possible ranges of $t_{cur}$. Once the optimum time difference profile $d(\tau)$, $0 \leq \tau \leq t_{cur}$ is obtained, the location of the current vehicle, vehicle B, can be obtained by examining the position of the reference vehicle, vehicle A, at the corresponding time. These formulations can be reformed in discrete forms as follows.

$$E_p = \sum_{i=0}^{P-1} |f_k - g_i|^2$$  \hspace{1cm} (14)

$$k = [i + d_i/\Delta d]$$  \hspace{1cm} (15)

$$i = 0, 1, \ldots, P - 1$$  \hspace{1cm} (16)

where $f$ and $g$ are the discrete versions of the altitude profiles of the reference and those of the evaluation. They are sampled at equally spaced time increments $t_i = i\Delta t$, where $\Delta t$ is the sampling interval, $i = 0, 1, \ldots, P - 1$, $k$ is the index of the reference altitude time series $f$, and $\Delta d$ is the step size of the time difference increment.

$[\ldots]$ in Equation (15) indicates the Gauss symbol which means the integer part function, that is, it returns the integer part of the argument. $P$ is the total number of pieces of time series altitude data. The sampling interval $\Delta t$ was set to 1.0 s in this study.

### 2.4. Yaw rate fusion of altitude and yaw rate

Yaw rate data is easily incorporated by modifying the measure function to take altitude and angular momentum (yaw rate) into account; that is, Equation (14) is replaced by the following equations.

$$E_p = \sum_{i=0}^{P-1} W_a |f_k - g_i|^2 + W_f |f y_k - g y_i|^2$$  \hspace{1cm} (17)

$$k = [i + d_i/\Delta d]$$  \hspace{1cm} (18)

$$i = 0, 1, \ldots, P - 1$$  \hspace{1cm} (19)

where $f_t$ and $g_t$ are altitude time series in m, $f y$ and $g y$ are the angular momentum, (yaw rate) in rad/s, and $W_a$ and $W_f$ are the weighting coefficients for altitude and angular momentum (yaw rate). $f$ means reference, and $g$ means data of the vehicle of interest.

The author simplified the problem by making the time difference $d(i)$ discrete values. In general, no vehicle can accelerate or decelerate too rapidly. That is, it can be assumed that the following relation holds.

$$d_{i+1} = \begin{cases} 
  d_i + \Delta d & \text{(acceleration)} \\
  d_i & \text{(cruse)} \\
  d_i - \Delta d & \text{(deceleration)}
\end{cases}$$  \hspace{1cm} (20)

where, again, $i$ is the index of the time series of evaluation (vehicle B) and $i = 0, 1, \ldots, P - 2$.

The selection of the value of $\Delta d$ depends on the resolution of the time interval. In this study, the time interval was set to 1.0 s, and $\Delta d$ was set to around 0.5 s assuming that the maximum acceleration and deceleration values of most moderate vehicles are around about 0.1 G. The author examined the results by changing the value of $\Delta d$ to around 0.5 s. If this quantization is too rough, it can be extended to have more variation in terms of velocity change. The value of $\Delta d$ was determined on the basis of the fact that the difference in the velocity of the two vehicles is at most 60 km/h, which means the headway will change at most 16.67 m within the time interval of 1 s.

### 2.5. Estimating position from time difference

If we denote the position vector of vehicle B as $pB(t_{cur}(i))$ and that of vehicle A as $pA(t_{ref}(i))$, where $t_{cur}(i)$ and $t_{ref}(i)$ are sampling times of the altitude time series data for vehicles B and A, respectively, and if we assume both vehicles A and B run the same route, the following equation holds.

$$pA(t_{ref}(i)) = (x_A(t_{ref}(i)), y_A(t_{ref}(i)))$$  \hspace{1cm} (21)

$$pB(t_{cur}(i)) = (x_B(t_{cur}(i)), y_B(t_{cur}(i)))$$
\[
\approx (x_A(t_{\text{cur}}(i) + d(i)), y_A(t_{\text{cur}}(i) + d(i)))
\]

(22)

\[
t_{\text{ref}}(i) = t_{\text{cur}}(i) + d(i)
\]

(23)

\[
i = 0, \ldots, P - 1
\]

(24)

From this, the position of vehicle B can be estimated by the position of vehicle A and the time difference between the two vehicles. If the position of vehicle A is measured precisely, the accuracy of the position of vehicle B is determined mainly by the accuracy of the estimated time difference.

3. Dynamic programming (DP) formulation of optimum estimation

To estimate the position of vehicle B (vehicle of 2nd run), the time difference profile must be estimated. Time difference profiles may vary depending on the situation because two vehicles may slow down or stop at intersections, and the speed is not always the same depending on the road and traffic conditions. The author assumes that if the time difference profile between vehicle A (vehicle of 1st run) and vehicle B is properly estimated, then the error in the altitude profiles of these vehicles of Equation (14) must be minimized. Hence, the author defines the following optimization problem, which can be formulated as a dynamic programming. In the process of the dynamic programming, the total amount of similarity matching error that is defined by Equation (14) is minimized by successive selection of \( d_i \), \( i = 0, 1, 2, \ldots, P - 1 \).

\[
d_{ij} = \left( j - \frac{Q - 1}{2} \right) \Delta d
\]

(25)

where \( i = 0, 1, \ldots, P - 1, j = 0, 1, \ldots, Q - 1 \), and \( Q \) is the number of discrete time difference values. In this study, \( Q = 201 \), that is, if the delay discrete step \( \Delta d \) is set to 0.5 s, the delay ranges from \(-50\) to \(50\) s at a resolution of 0.5 s. These nodes, which represent the time difference candidates at each time index, are illustrated in Figure 2. Each time difference node in Figure 2 has a measure \( F_{i+1,j} \) that reflects the preceding choice of time difference nodes as follows.

\[
F_{i+1,j} = \min \left\{ F_{i,j-1} + E_{i+1,j-1,j}, F_{i,j} + E_{i+1,j+1,j}, F_{i,j+1} + E_{i+1,j+1,j} \right\}
\]

acceleration (cruise) (deceleration),

(26)

where \( i \) is the time index, \( j \) is the discrete time difference index, \( i = 0, 1, \ldots, P - 1 \), and \( j = 0, 1, \ldots, Q - 1 \).

\( E_{i+1,j,j'} \) is the similarity matching error at time index \( i + 1 \) if the time difference changes from \( j' \) at time index \( i \) to \( j \) at time index \( i + 1 \). The local optimum time difference \( d_{\text{opt}} \) that satisfies Equation (26). That is, if the current time difference is \( d_i \), and if the optimum choice of \( d_j \) at previous stage \( i \) out of three candidates \( \{d_{i,j-1}, d_{i,j}, d_{i,j+1}\} \) is \( d_{\text{opt}} \), the time difference at

**Figure 2.** Lattice network for estimating optimum delay profile. This can be solved by dynamic programming. Minimization of measure can be performed step by step.
time index \( i + 1 \) is given by this value:

\[
d_{i+1,j} = d_{i,j^{\text{opt}}},
\]

where \( d_{i,j^{\text{opt}}} \) is the time difference corresponding to \( F_{i,j^{\text{opt}}} \) that has been obtained in advance also on the basis of Equation (26) at an earlier stage \( i \). The altitude reference data \( f_k \) is the data at \( t = i \Delta t + d_{i,j^{\text{opt}}} \), that is, \( k = i + j^{\text{opt}} \) in discrete form representation of Equation (14) and is compared with the current altitude data as follows.

\[
E_{i,j^{\text{opt}}} = |f_i + j^{\text{opt}} - g_i|^2
\]

The initial values of \( F_{0,j} \), \( j = 0, \ldots, Q - 1 \) are given as follows.

\[
F_{0,0} = |f_0 - g_0|^2
\]

\[
F_{0,j} = \infty, \quad j = 1, \ldots, Q - 1
\]

Equations (26), (27), and (28) indicate that vehicle B starts from the origin at time index \( i = 0 \) with velocity 0, and the velocity estimation assumes that vehicle B is still before the start and no velocity nodes with respect to \( j = 1, 2, \ldots, Q - 1 \) are included in the candidates at time index \( i = 0 \). Each node corresponding to \( F_{i,j} \) has a discrete time index \( i \) and delay index \( j \).

The optimum sequence of the time difference can be calculated by applying these equations repeatedly from the first index \( i = 0 \) to the current time index. This is propagating values over a lattice structure as is shown in Figure 2. The calculation is done according to the elapsed time and time difference between the two vehicles axes, and the author calls this forward chaining estimation of the time difference profile which is suitable for real-time location estimation. In this case, the algorithm finds the minimum error path in a lattice network at all times, which gives the minimum error at the current time index. If the real-time constraint is not strict, backward chaining estimation is possible, which is, in general, more reliable. Backward chaining estimation starts from the current time index, reviews the result of the forward chaining, and follows the minimum error path backwards to the beginning time index \( i = 0 \). The order of computation is \( O(PQ) \), where \( P \) is the number of pieces of time series data, and \( Q \) is the number of discrete delays. If this algorithm is run in a real-time situation, the order is \( O(Q) \) at each time interval.

4. Experimental results

The proposed algorithm was evaluated by applying atmospheric data and angular momentum data acquired on arterial roads from Yoshioka hot spring and Shikano hot spring (Shikano-Ourai Road, prefectural road No. 21) in Tottori, Japan, as described in Figure 3 and Table 1. The total length of the test course the author chose was about 9.8 km. Field test courses should incorporate different types of roads to evaluate the algorithm fairly. The test course includes rugged sections where the altitude changes significantly (a gradient of about 8%, which is the upper limit of Japanese trunk road construction regulations) and sections where the rises in elevation per 100 m are less than 0.1 m, that is, a gradient of 0.1%. Therefore, the test course contains various situations with altitude changes, so the author thought that no other test courses would be necessary to evaluate the proposed algorithm at this time.

4.1. Altitude time series data obtained by vehicle

By using an atmospheric sensor embedded in a smartphone, a time series of pressure can be obtained. As mentioned above, the author measured atmospheric data along a road called Shikano Ourai Road, which is 9.8 km long and runs between Yoshioka hot spring and Shikano hot spring near our university. The trajectory of the GNSS data is shown in Figure 3. To confirm the data acquisition.

| Date          | September 5, 6, 2020       |
|---------------|----------------------------|
| Weather       | clear (Sept. 5), cloudy (Sept. 6) |
| Temperature   | 29 degrees Celsius both days |
| Vehicle       | Daihatsu Mira Geno (660 cc) |
| Atmospheric sensor | Bosch BMP380 (50 Hz) |
| Gyroscope     | Bosch BMI160 (50 Hz)        |
| GNSS receiver | u-Blox ZED-F9P (5 Hz)       |
| Trip length   | 9.8 km, one-way course      |
| Running time (1st run) | 735 s (from 10:49:03–11:01:18) |
| Running time (2nd run) | 666 s (from 11:43:10–11:54:16) |
| Origin        | Yoshioka hot spring        |
| Destination   | Shikano hot spring         |
feasibility of the proposed localization algorithm, the author obtained reference altitude and angular momentum (yaw rate) time series and those of evaluation time series data. A real-time kinematic (RTK)-GNSS receiver U-Blox ZED-F9P [15], was installed on the rooftop of a mini vehicle at the time of collecting the atmospheric pressure and angular momentum (yaw rate) data. The RTK-GNSS latitude and longitude calculation interval was set to 0.2 s. Fortunately, the RTK-GNSS wave reception had no problem, and ground truth data was successfully obtained by the RTK-GNSS on this test course. Two trips were performed.

Two atmospheric pressure time series are shown in Figure 4. The data was obtained on different days, so the baseline of the atmospheric pressure was different. From this data, two altitude time series were obtained with their start points set to the same location (Yoshioka hot spring) and their origin of time both set to \( t = 0 \). The two altitude time series are shown in Figure 5. The experimental conditions are shown in Table 1. It can be seen that the altitude data obtained from this result agreed quite well as shown in Figure 5 even though a slight drift that may have been due to the sensor itself or from a change in the weather was observed, and also a shift in travel times was observed due to a stop of the reference run vehicle at an intersection at about 150 s on the time axis due to a red traffic light, while the vehicle of interest did not have to stop at the intersection because the traffic light was green.

The MEMS pressure sensor used was a Bosch BMP380 [16] whose absolute accuracy is \( \pm 0.5 \) hPa and relative accuracy is \( \pm 0.06 \) hPa. This is equivalent to roughly an absolute error of \( \pm 4 \) m and relative error of \( \pm 50 \) cm [14]. The author examined the actual fluctuation of the atmospheric sensor output by calculating the standard deviation of pressure data obtained by actually measuring at a fixed point for 5 s. The atmospheric sensor sampling frequency was 50 Hz. As a result, the standard deviation of the atmospheric pressure data was 0.018 hPa, which is 15.6 cm in altitude. The altitude profile time series obtained by the pressure sensor was used as is, and all processing was done in the time domain in this paper. Hence, transformation of the time series to a spatial altitude data series was not necessary. The altitude time series data was obtained by the pressure sensor while driving the vehicle. Because of the change in velocity during driving, the altitude profile in the time series was deformed along the time axis. Suppose we have a reference altitude profile \( f(t) \) along a route and precise location \( x_A(t) \) that was obtained by vehicle A. Next, we obtain a new altitude profile \( g(t) \) along the same route with another vehicle B, and our target is to determine the location \( x_B(t) \) of vehicle B assuming that vehicle B is not equipped with GNSS. Here, \( f(t) \) represents the reference time series of the altitude obtained by vehicle A, and \( g(t) \) represents the time series of the altitude under evaluation.

### 4.2. Yaw rate time series data obtained by a vehicle

As another information source, the author thought that angular momentum, especially yaw rate, could be a candidate for the feature of roads because left and right turn manoeuvrers and driving along curves will be reflected in the yaw rate efficiently compared with pitch and roll. Therefore, in this paper, yaw rate data is incorporated in the same manner as the altitude data. Figures 6 and 7 show the yaw rate of the reference vehicle and vehicle of interest. Similarity is clearly observed between these two profiles. Positive values mean left turn and negative values mean right turn, and the yaw rate clearly reflects the characteristics of the road.

The gyroscope used was a Bosch BMI160, and the sampling frequency was 50 Hz [17].
4.3. Effectiveness of sensor fusion of altitude and yaw rate

The time difference profile estimated by altitude profile is shown in Figure 8. The time difference increment parameter was $\Delta d = 0.5$ s. The solid brown line is the time difference profile estimated by the backward chaining algorithm based on altitude data. The result of forward chaining is shown with the dotted brown line for comparison. The GNSS-measured ground truth of the time difference is also shown by the solid blue line for comparison. The estimated time difference became close to the ground truth in the range roughly from 100 to 400 s.

The time difference profile estimated by yaw rate profile is shown in Figure 9. The solid brown line is the time difference profile estimated by the backward chaining proposed algorithm based on yaw rate data. Again, the GNSS-measured ground truth is also shown by a solid blue line for comparison. The estimated time difference also became close to the GNSS-measured delay in the range roughly from 100 to 450 s. It is surprisingly comparable to the result obtained by altitude similarity matching in Figure 8. It is clear that angular momentum (yaw rate) has a comparable information as does the altitude data.

Figure 10 shows the results of location error. The result of sensor fusion, which is plotted in green, was the best result. It was almost within 20 m in the range of 100–450 s. However, in another region, the error increased. In the region of 0–100 s, a significant location error occurred due to an unwanted stop of the reference vehicle due to a traffic light. The vehicle of interest did not stop there, so it overpassed the reference vehicle. The time difference increased rapidly, and the DP algorithm took a long time to make up for the difference, resulting in a significant location error. The large error in the range of about 450–650 s was still due to the little variation and little change in the yaw rate. Introducing angular momentum (yaw rate) did not help much in this region.

4.4. Effectiveness of eliminating unwanted vehicle stops from reference time series data

From the results of the previous subsection, it was shown that if the vehicle acquiring the reference data had to make a stop and the vehicle under evaluation did not stop at that particular point, a rapid time difference will occur that cannot be compensated for with the DP algorithm, and making up for this difference...
takes a long time, resulting in large position errors such as shown in the range before 100 m in Figure 10. This may be overcome by introducing more candidates for the time difference in Equation (20). At this time, the number of candidates is limited to 3 to make the DP algorithm simple. Increasing the number of candidates directly increases the processing time, so the author chose another counter measure. The measure is quite simple. It eliminates unwanted stops by skipping the length of time that elapsed when the reference vehicle stopped at a particular position.

Figure 11 shows a velocity profile of the reference vehicle obtained by wheel pulse data when it stopped at an intersection shortly after it started from the start point. If we set a threshold value such as 2.0 km/h for deciding whether the vehicle is stopped, it can be seen that it stopped for 25 s from 23 to 48 s at a traffic light. Thus, unwanted stops in the reference data can be extracted and the reference data can be modified.

Figure 12 shows the result of time difference estimation by altitude data using the modified reference data. From the effect of eliminating unwanted stops from the reference time series data, the errors shortly after the start from 0 to 100 s were greatly reduced compared with Figure 8. However, the errors in the range of 400–650 s were, again, large.

Figure 12. Result of time difference estimation based on dynamic programming and its comparison with that obtained by RTK-GNSS. Only altitude data was used. Time difference was estimated by backward chaining of DP sequence.

Figure 13. Result of time difference estimation based on dynamic programming and its comparison with that obtained by RTK-GNSS. Only angular momentum data was used. Time difference was estimated by backward chaining of DP sequence.

Figure 13 shows the result of time difference estimation by yaw rate data using the modified reference data. From the effect of eliminating unwanted stops from the reference time series data, the errors shortly after the start from 0 to 100 s were greatly reduced compared with Figure 9. However, the errors in the range of 450–500 s were still large even though the errors in the range after 500 s decreased from Figure 9. This result shows the effectiveness of yaw rate data.

Figure 14 shows the result of time difference estimation with the sensor fusion algorithm using the modified reference data. Both the altitude data and yaw rate data were used in the evaluation function defined by Equation (17)–(19). From the effect of eliminating unwanted stops from the reference time series data, the errors shortly after the start from 0 to 100 s were greatly reduced compared with Figure 15. Also, from the effect of sensor fusion, errors in most all of the ranges decreased, with errors still existing in the range of 450–500 s.

Figure 16 shows the result of vehicle location estimation error by the sensor fusion algorithm after the reference data was modified to eliminate the unwanted stop. The error due to the unwanted stop was greatly reduced in the region before 100 s in this case. Among the three, the result obtained by the sensor fusion was the best. In most of the regions, the location error was below 20 m;
Figure 14. Result of time difference estimation based on dynamic programming and its comparison with that obtained by RTK-GNSS. Both altitude and angular momentum data were used. Parameters were $W_a = 1.0$ and $W_y = 2 \times 10^5$. Time difference was estimated by backward chaining of DP sequence.

Figure 15. Result of time difference estimation based on dynamic programming and its comparison with that obtained by RTK-GNSS. Both altitude and angular momentum data (yaw rate) were used. Parameters were $W_a = 1.0$ and $W_y = 2 \times 10^5$. Time difference was estimated by backward chaining of DP sequence.

Figure 16. Location errors between estimated and ground truth data of reference vehicle. Time differences are estimated by backward chaining of DP. Only the result of sensor fusion is shown. Figure 14 corresponds to this result.

However, there was still an error that exceeded 100 m for a region with little variation in altitude and yaw rate.

4.5. Discussion

Looking at Figures 8, 9, and 15, it is remarkable that the result of estimation based on the sensor fusion (Figure 15) was better than that of only altitude data (Figure 8) or only yaw rate data (Figure 9). After eliminating the unwanted stops, this also applied. The result of estimation based on the sensor fusion (Figure 14) was better than that of only altitude data (Figure 12) or only yaw rate data (Figure 13).

However, the sensor fusion of altitude and yaw rate could not improve the localization accuracy for regions with a small altitude variation and also small yaw rate change such as in the region of 550–600 s.

Figure 17 shows the road gradient profile of the test course along the time axis. This profile was obtained from altitude data obtained with a RTK-GNSS receiver, the U-Blox ZED-F9P, equipped in the 2nd run for evaluation. The RTK-GNSS has a sufficient enough accuracy of 0.01 m.

Figure 18 shows the road gradient profile along the distance of the test course.

From Figure 17, it can be seen that there are several sections with a gradient of almost less than 2%, such as in the region of about 420–650 s, which is about 6000–9800 m in distance from the start. The variation in yaw rate is also small in this region as can be seen from the yaw rate data of Figure 7.

In Figure 3, the GNSS trace shows an almost straight line in the south west part of the trip, and very little yaw rate change was observed. The largest error of the estimated time difference was about 6 s which corresponds to a position error of about 120 m. For these sections, other features such as pitch rate, roll rate, and acceleration for several directions should be incorporated and are under study. Wheel pulse signals are also a candidate for an additional information source for dead-reckoning even though their accumulative error when estimating the distance travelled by multiplying...
a coefficient with the number of pulses must be taken into account.

5. Conclusion

A new algorithm for localization based on altitude and angular momentum information was proposed. Altitude and angular momentum (yaw rate) reference data is obtained with altitude probes on vehicles. Atmospheric pressure data can be used to acquire altitude data. Time series of altitude data are directly processed in the time domain. From the results of an evaluation, the following findings were obtained.

- A similarity matching algorithm was formulated under a dynamic programming paradigm. A lattice network representation is useful for the problem in which each node represents a discrete time difference between the reference vehicle and the vehicle under evaluation.
- Both a forward chaining and backward chaining algorithm were developed. The forward chaining algorithm is suitable for real-time localization. However, the backward chaining algorithm is more stable and gives better results.
- If there is little variation in the altitude and yaw rate profile, the accuracy of position estimation is degraded. For those cases, we must incorporate other information such as acceleration data or wheel pulse data for better results.
- The proposed algorithm was applied to actual altitude data and angular momentum (yaw rate) data to examine its potential feasibility. As a next step, there should be more theoretical and quantitative study on how spatial variation in altitude and the accuracy of sensors influence the accuracy of location estimation based on altitude and yaw rate profiles. This is the author’s future work.
- The result obtained by the sensor fusion of altitude and yaw rate outperformed the results for altitude only or yaw rate only.
- In this study, parameters such as the time difference increment parameter were set around 0.5 s, the time resolution was set to 1.0 s, and the number of candidates of discrete time difference was set to three cases that represent accelerate, cruise, and decelerate cases for simplicity. However, there do not have to be three cases, and having more candidates may increase the accuracy of the estimation. Because, these parameters influence the estimation result, they need to be examined by testing more cases and should be selected, if possible, from more theoretical viewpoints.
- In this study, a fixed route was used for evaluation. It must be extended to a general road network with many candidate road links.

Acknowledgments

This research was supported by JSPS KAKENHI Grant Number 20K04723.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was supported by Japan Society for the Promotion of Science (JSPS) KAKENHI [grant number 20K04723].

Notes on contributor

Takayoshi Yokota has been a professor at Tottori University since April 2012. He received his B.S., M.S., and Ph.D from Tokyo Institute of Technology in 1979, 1981, and 1984. He joined the Hitachi Research Laboratory of Hitachi, Ltd. in 1984. He was a professor in the faculty of Kyoto University in April 2009. He has been a professor of the Graduate School of Engineering at Tottori University since April 2012. His current research interests include analysis, modelling, and spatio-temporal information processing of Intelligent Transport Systems (ITS) related data. He was awarded a best scientific paper award at the 26th ITS World Congress, 2019. He is also a member of IEEE, IEICE, and IPSJ.

ORCID

Takayoshi Yokota http://orcid.org/0000-0002-3229-2250

References

[1] Yokota T. Localization algorithm based on altitude time series in GNSS-denied environment. In: 2020 Proceedings of the SICE Annual Conference; Chiang Mai, Thailand; 2020 Sep 23–26. p. 952–957.
[2] Yokota T, Okude M, Sakamoto T, et al. Fast and robust map-matching algorithm based on a global measure and dynamic programming for sparse probe data. IET Intell Transp Syst. 2019 Nov;13(11):1613–1623.
[3] Yokota T, Tamagawa D. Constructing two-layered freight traffic network model from truck probe data. Int J Intell Transp Syst Res. 2011 Jan;9(1):1–11.
[4] Tsurushiro J, Nagaosa T. Vehicle localization using its vibration caused by road surface roughness. In: Proceedings of the 2015 IEEE International Conference on Vehicular Electronics and Safety; Yokohama, Japan; 2015 Nov. p. 164–169.
[5] Tsurushiro J, Nagaosa T, Mizui K. A study of vehicle localization based on road surface roughness using sensor data in a smart phone. 2017 Feb. p. 57–62. (IEICE technical report; ITS2016–86).
[6] Dean AJ, Martini RD, Brennan SN. Terrain based road vehicle localization using particle filters. In: Proceedings of the 2008 American Control Conference; 2008 Jun. p. 236–241.
[7] Dean AJ, Martini RD, Brennan SN. Terrain based road vehicle localization on multi-lane highways. In: Proceedings of the 2009 American Control Conference; 2009 Jun. p. 707–712.
[8] Dean AJ, Martini RD, Brennan SN. Terrain-based road vehicle localisation using particle filters. Int J Veh Mech Mob. 2011;49(8):1209–1223.

[9] Laffchiev E, Lagoa C, Brennan S. Terrain-based vehicle localization from real-time data using dynamical models. In: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC); ACC, Maui (HI) USA; 2012.

[10] Gim J, Ahn C. Ground feature-based vehicle positioning. In: Proceedings of the SICE Annual Conference; 2020. p. 983–984.

[11] Gim J, Ahn C. IMU-based virtual road profile sensor for vehicle localization. Sensors. 2018;18(10):33–44.

[12] Qu X, Soheilian B, Paparoditis N. Landmark based localization in urban environment. ISPRS J Photogramm Remote Sens. 2019 Jun;140:90–103.

[13] Agrawal and M, Konolige K. Real-time localization in outdoor environments using stereo vision and inexpensive GNSS. In: 18th International Conference on Pattern Recognition; ICPR 2006. IEEE; 2006. p. 1063–1068.

[14] [accessed on 2020 Dec 24]. Available from: https://ntrs.nasa.gov/api/citations/19770009539/downloads/19770009539.pdf.

[15] [accessed on 2020 Dec 24]. Available from: https://www.u-blox.com/en/product/zed-f9p-module.

[16] [accessed on 2020 Dec 24]. Available from: https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp380/.

[17] [accessed on 2020 Dec 24]. Available from: https://www.bosch-sensortec.com/products/motion-sensors/imu/bmi160.html.