A Map Matching Algorithm for Car Navigation Systems that Predict User Destination

Koichi Miyashita, Tsutomu Terada, and Shojiro Nishio
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
{miyashita.koichi, tsutomu, nishio}@ist.osaka-u.ac.jp

Abstract

In this paper, we propose a map matching algorithm for car navigation systems that predict user destination. This car navigation system is a novel system that automatically predicts user purpose and destination to present various information based on predicted purpose without user interaction. It requires correct road links where the car drives in real time, and it also need to know the route from start point to current point correctly. The proposal method achieves a real time map matching that can create a correct route by using the enhanced shortest path algorithm. The simulation study confirmed the effectiveness of our method.

1 Introduction

With advances in computer miniaturization and information technologies, car navigation systems have come into widespread use as useful tools that support drivers by navigating them where they want to go. Car navigation systems are approaching the saturation of navigation functions in response to destinations input into the system. On the other hand, users do not usually input them into the system because familiar routes are used for daily driving and they do not need any routing information to get to the destination. This means that we do not exploit the full effectiveness of car navigation systems in daily life. We require an useful car navigation system without stressful work in non-navigating situations. In response to this requirement, we propose a new car navigation system that automatically presents location-aware information by predicting driver purpose and destinations[5].

To predict driver's destination, our system and other destination predicting algorithms need accurate trip histories that a series of roads the driver ran. However, a raw data of trip histories includes a lot of errors. For example, there are a measurement error margin in the Global Positioning System (GPS) and cumulative errors in the dead reckoning method using various sensors. Because of these noises, the system cannot predict driver's destination accurately. In this paper, we propose a new method to remove these noises and to get accurate trip histories, and we evaluate our method by comparing to right histories.

The reminder of this paper is organized as follows. In Section 2, we present our car navigation system that predicts user destination. In Section 3, we introduce some map matching methods. Section 4 describes our map matching method. In Section 5, we evaluate our proposed algorithm. Finally, we summarize this paper in Section 6.

2 Car Navigation System That Predicts User Destination

2.1 Service Scenarios

Our system automatically shows information suitable for user's destination which is predicted by the system. The followings are examples of service scenarios:

- It automatically predicts that the user destination is a shopping mall and presents information of another parking lot because the route to the usually used parking lot is now congested with traffic.
- It automatically recognizes that the driver is taking friend to the station and presents a train schedule for an estimated arrival time.
- It detects the possibility of running out of gas, alerts the driver, and shows information about gas stations along the route to the destination.

A snapshot of our implemented prototype system is shown in Figure 1. Our system uses several animation characters which show information based on the destination prediction.

2.2 Destination Prediction

In the destination predicting algorithm that our research group proposed, the present driving route from start position to current position is compared with the trip histories of past drives, and the destination of the running history

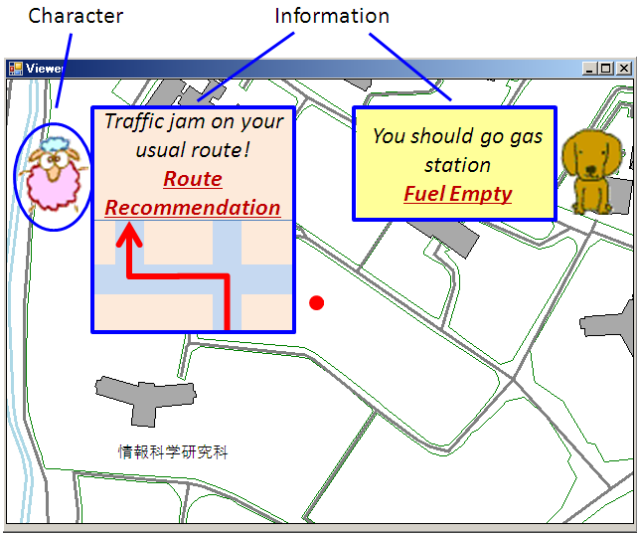

Figure 1. Prototype image

that is similar to a present running trajectory is extracted as the destination where the user is heading now.

The system records the transition history of road links, and each road link also stores the frequency of going to each destination by the link. Specifically, suppose $L = (l_0, l_1, l_2, ..., l_i)$ represents the current transition history of road links, and the user is now on link l_i. In this situation, probability $P_{l_i d}$ that he/she at link l_i goes to destination d is expressed as the following formula:

$$P_{l_i d} = (1-\alpha)\frac{N_{l_i d}}{N_{l_i}} + \alpha P_{l_{i-1} d} \qquad (2.1)$$

In the formula, N_{l_i} means a total frequency that has passed road link l_i before, $N_{l_i d}$ means a total frequency where he has gone to destination d through road link l_i, α is a coefficient to control the weight of considering past routes for calculating$P_{l_i d}$ and the value from 0 to 1 is taken. Note that, the meaning of $P_{l_0 d}$ is the ratio of a total frequency where the driver has gone to destination d to a total number of drives. When the vehicle comes into next link, our system calculates this formula. For example, P_{*A} is calculated as shown in Figure 2. In this way ($O \rightarrow a \rightarrow e \rightarrow A$), the system predicts the possibility of visiting destinations using route record (Table. 1).

3 Map Matching

The map matching is a technique to match the provisional present location, that obtained by GPS and some sensors, to the road on the map, by using the rule that vehicles usually run on a road. This map matching function is needed in not only an existing car navigation system but also the system that predicts a driver's destination and the system that predicts future traffic flow.

Table 1. Route record

Route	Count
$O \rightarrow b \rightarrow B$	6
$O \rightarrow a \rightarrow c \rightarrow C$	3
$O \rightarrow a \rightarrow e \rightarrow A$	2
$O \rightarrow d \rightarrow c \rightarrow C$	5
$O \rightarrow d \rightarrow e \rightarrow A$	4

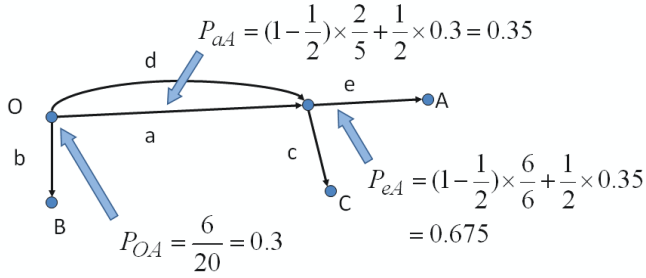

Figure 2. Prediction example (α = 0.5)

Especially, it is essential in our system described in Section 2 and other driver's destination prediction methods such as a method proposed in [4]. Therefore, correct running trajectories are important for these algorithms to calculate right probabilities and to predict correct destinations.

3.1 Map Matching In Current Navigation System

A position specification technique of present car navigation systems combines Global Positioning System (GPS) and Dead Reckoning to specify the current position. Dead Reckoning is the process of estimating one's current position based on a previously determined position, and advancing that position based on several condition information acquired by various sensors (speed, acceleration, running distance). Since the dead reckoning accumulates the error margin with the passage of time, GPS is used for preventing this error accumulation. The system retrieves the road links that are near the previous position, and calculates similarity between the road links to the excursion gotten by GPS and dead reckoning method in certain intervals. Then, the current location is matched to the road link that has highest similarity. Since this matching method has a feature of matching a present vehicle position to a plausible road, it is suitable for the car navigation.

However, this matching method has a problem of generating an unnatural jump in tracks of the route when a point is matched incorrect road. For example, when a vehicle approaches Y junction (Figure 3. (a)) and present location (P_1) is matched to wrong road R_2 by calculation result of similarity of P_1 and around road links, following GPS points are matched to roads that connect to R_2 (Fig-

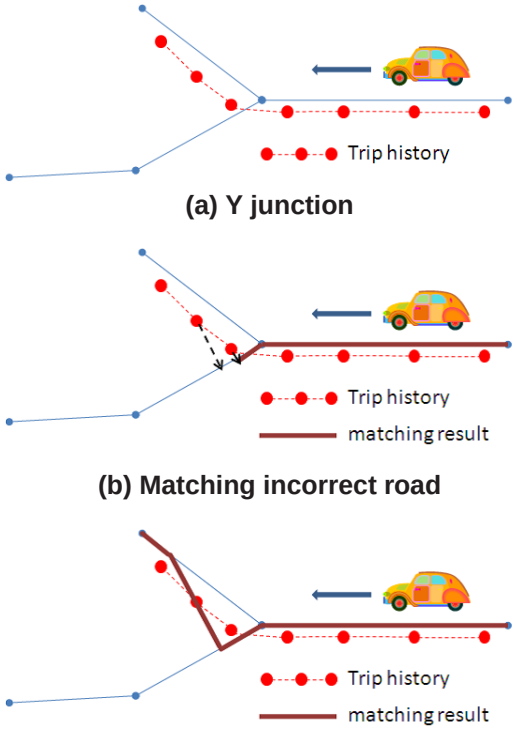

Figure 3. Incorrect map matching

ure 3. (b)). After that, when the vehicle approaches P_3 and matching target roads that connect to R_2 is far from P_3, its position is matched to R_3 that is correct running road. In this case, "road to road jump" occurs (Figure 3. (c)).

3.2 Related Work

Researches on map matching can be divided into two categories; real time map matching and static map matching.

3.2.1 Static Map Matching

Static map matching is a method that returns a trajectory from start point to end point after the end of drive. Brakatsoulas[1] introduces a map matching algorithm bases on *Fréchet Distance*. A popular illustration of *Fréchet Distance* is the following: Suppose a person is walking his dog, the person is walking on the one curve and the dog on the other. Both are allowed to control their speed but they are not allowed to go backwards. Then, *Fréchet Distance* of the curves is the minimal length of a leash that is necessary for both to walk the curves from beginning to end. This research proposes a method to find a route whose *Fréchet Distance* is lower than any other route. If this method is used in a real time environment, the route from start position to current position will be calculated again at each update of the location information. Therefore, it is not suitable for real time processing.

3.2.2 Real Time Map Matching

Real time map matching is a method that returns a trajectory from start point to current point in real time. Yang[3] introduces a method for real time map matching. He proposes a technique for matching GPS points that are received in each two minutes. This environmental assumption differs from ours in that GPS location can be acquired in per second. Because of acquiring vehicle position at low frequency, this method cannot extract right route but only likely route.

Brakatsoulas[1] and Greenfeld[2] introduce a real time map matching with high frequency location updating. First, it searches a road link that is the nearest from start GPS position. Next, the road links connected with the road link matched ahead are compared with a present position. And, this processing is carried out at each update of location information gotten by GPS. It calculates the similarity of the line to the roads based on the distance and the angle of two lines, and matches current position to most similar road. In Brakatsoulas's method, the distance parameter S_d is defined as formula 3.1, and the angle parameter S_α is defined as formula 3.2. In these formulas, p_i is current position, c_j is a road link that is a candidate to match p_i, $d(p_i, c_j)$ is the function to get a length between p_i and c_j, $\alpha_{i,j}$ is the angle between p_i and c_j, and the score of c_j is given by $S_d + S_\alpha$. The road link that has highest score is selected to match p_i. $\mu_d, a, n_d, \mu_\alpha$, and n_α are scaling factors and the authors empirically established the following parameter settings $\mu_d = 10$, $a = 0.17$, $n_d = 1.4$, $\mu_\alpha = 10$, and $n_\alpha = 4$.

$$S_d(p_i, c_j) = \mu_d - a \cdot d(p_i, c_j)^{n_d} \tag{3.1}$$

$$S_\alpha(p_i, c_j) = \mu_\alpha \cdot \cos(\alpha_{i,j})^{n_\alpha} \tag{3.2}$$

This method has a weak point of receiving the influence of a noise easily to repeat this processing every point. Additionally, there is a problem of dragging the mistake because it is matching roads that connect to the incorrect road after mismatching. Specially, if the matching mistake occurred at beginning point of driving, it is likely to keep matching to incorrect roads for a long term.

4 Proposed Algorithm

Our proposed map matching method extends the shortest path algorithm. *Shortest path method* that extends the shortest path algorithm is a basic of this map matching algorithm, and *Divided shortest path method* is an improved version of *Shortest path method*.

4.1 Shortest Path Method

This method treats a road network as a network and intersections as network nodes, and road links between two intersections as network edges as well as other matching

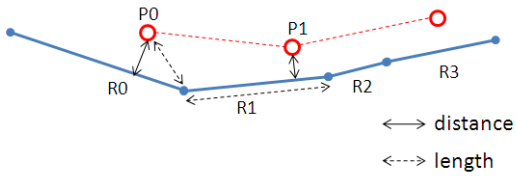

Figure 4. Example of scoring

techniques. A road link does not connect with other road links excluding the intersection nodes.

The shortest path algorithm is an algorithm that adds an arbitrary score to the link that composes the network, calculates the sum of the score of links where by way of in the route that connects the starting point with the terminal point, and finds the route to which this total score is minimized. The score putting on the road link in this paper is defined as formula 4.1.

$$Score = length \times distance \tag{4.1}$$

$length$ is a length of a road link, and $distance$ is a distance between a running trajectory and the road link. Figure 4 illustrates the examples of $length$ and $distance$ for each road link. Note that, in start point and end point of trajectory, $length$ is the distance from the point to the edge point of the target road link. For example, in Figure 4, start point $P0$ is near edge point of road link $R0$. In this case, if the length of $R0$ defined $length$, the score of $R0$ becomes too large and $P0$ is matched $R1$. To avoid this case, it takes this exception.

Finally, the route that connects the start point to the terminal road is extracted by the shortest path algorithm, and a minimum score route becomes a correct answer route.

4.2 Divided Shortest Path Method

Some considerable map matching error occurs in several situations though most running trajectories can be matched correctly by using the basic algorithm previously described. For example, there are situations that a driver runs in the circuit (Figure 5), that a driver generates the turn while running (Figure 6), and that a driver takes a roundabout course (Figure 7). In such cases, the characteristic of the shortest path algorithm appears strongly and it is not possible to match to the correct road links. The reason why the problem appears is the use of shortest path algorithm for all road links and all trajectories at a time. Moreover, a method that matches a whole trajectory at a time is unsuitable for a real time matching.

To solve these problems, we propose *Divided Shortest Path Method*. This method divides the trajectories into constant sections, and executes the base algorithm with small divided sections.

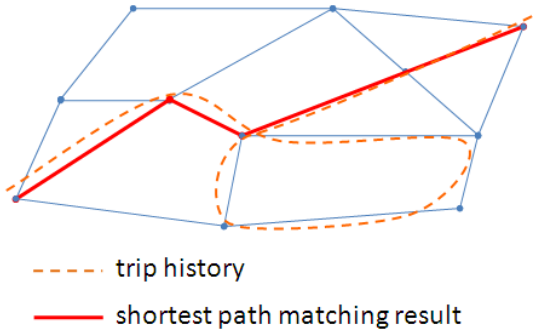

Figure 5. Example of wrong matching(1)

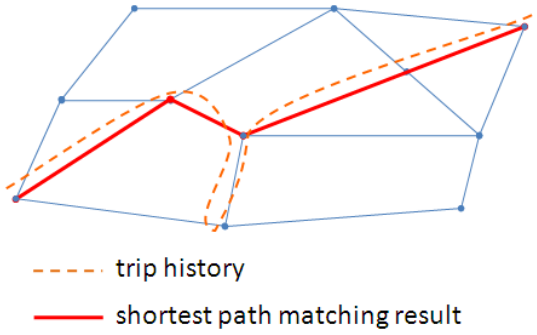

Figure 6. Example of wrong matching(2)

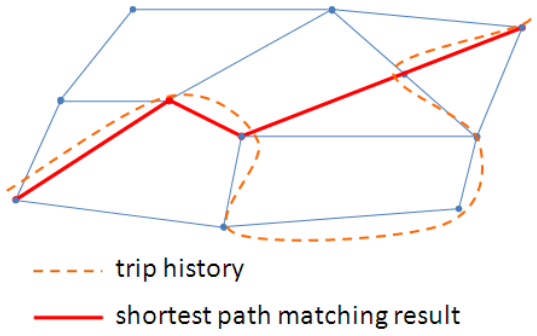

Figure 7. Example of wrong matching(3)

To be concrete, provisional vehicle position G_i is extracted at regular intervals. G_i is a latest selected point and G_{i-1} is a previous selected point. It retrieves J pieces of road links $R_{G_ij'}$, $(j' = 1, 2..J)$. Here, J is a number of road links that are positioned within $threshold$ from G_i. For instance, supporting a case where in Figure 8. The road links R_{G_41}, R_{G_42}, R_{G_43} are extracted for.

By using the shortest path method, it calculates $S(R_{G_{i-1}j''}, R_{G_ij'})$ that is a score of the routes from road links $R_{G_{i-1}j''}$, $(j'' = 1, 2..J)$ in surroundings of previous

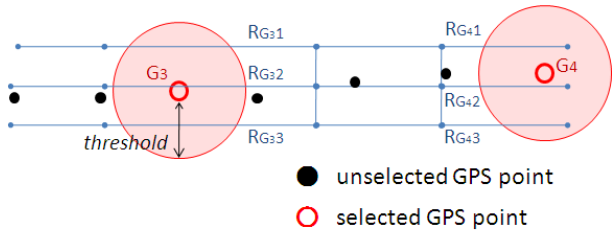

Figure 8. Example of divided shortest path method

Table 2. $S(R_{G_3j}, R_{G_4j})$

	R_{G_41}	R_{G_42}	R_{G_43}
R_{G_31}	300	200	900
R_{G_32}	300	100	500
R_{G_33}	700	300	200

Table 3. TS_{G_3j}

	Total Score
TS_{G_31}	1000
TS_{G_32}	500
TS_{G_33}	700

point G_{i-1} to road links $R_{G_ij'}$.

Next, defining an integrated score of the route from R_{G_0j} to $R_{G_{i-1}j''}$ as $TS_{G_{i-1}j''}$, integrated score of the route from road link R_{G_0j} to $R_{G_ij'}$ is obtained from the following formula 4.2.

$$TS_{G_ij'} = min(TS_{G_{i-1}j''} + S(R_{G_{i-1}j''}, R_{G_ij'})) \quad (j'' = 1, 2..J) \tag{4.2}$$

Road link $R_{G_ij'}$ where an integrated score $TS_{G_ij'}$ is the smallest is adjusted to a provisional vehicle position of present place G_i (formula 4.3).

$$G_i = min(TS_{G_ij'}),\ (j' = 1, 2..J) \tag{4.3}$$

In the case of Figure 8, $S(R_{G_3j}, R_{G_4j})$ that is a score of shortest path method from road links R_{G_3j} in surroundings of previous point G_3 to R_{G_4j} is shown in Table 2. And, integrated score TR_{G_3j} of the route from road link R_{G_0j} to R_{G_3j} is shown in Table 3. In this case, by using $TS_{G_ij'} = TS_{G_{i-1}j''} + S(R_{G_{i-1}j''}, R_{G_ij'})$, the result is shown in Table 4. Lastly, it calculates a integrated score TR_{G_4j} (Table 5) and determines the matched road link. In this example, road link R_{G_42} that has the smallest integrated score is selected to match G_4. Then it can make a route from the start point to G_4 by tracking like $R_{G_42} \rightarrow R_{G_32} \rightarrow \cdots \rightarrow R_{G_0j}$.

5 Evaluation

5.1 Evaluation Environment

To evaluate the accuracy of our map matching algorithm, we collected GPS log data for three months (total driving time: 20 hours, number of driving: 57, average driving time: 20 minutes) using Garmin GPS eTrex VIsta-C.

Table 4. $TS_{G_3j} + S(R_{G_3j}, R_{G_4j})$

	R_{G_41}	R_{G_42}	R_{G_43}
R_{G_31}	1300	1200	1900
R_{G_32}	800	600	1000
R_{G_33}	1400	1000	900

Table 5. TS_{G_4j}

	Total Score
TS_{G_41}	800
TS_{G_42}	600
TS_{G_43}	900

Table 6. Rate of concordance

	Same rate	Over rate	Lack rate
shortest path	88.74	1.90	9.36
divide (10 minutes)	90.18	2.52	7.30
divide (7 minutes)	91.56	2.54	5.90
divide (5 minutes)	94.92	2.21	2.87
divide (3 minutes)	94.00	3.09	2.91
divide (1 minute)	93.58	3.54	2.87
divide (1 second)	84.91	9.91	9.91
simple real time	41.30	9.15	49.55

These drive trajectories were matched to the Japanese digital map of 1/2500 scale that Japanese Government distributes. We compare three methods, Brakatsoulas's real time map matching method (cf. Section 2.2) that we call *simple real time method* , *shortest path method* (cf. Section 3.1), and *divided shortest path method* (cf. Section 3.2). In *divided shortest path method*, the division interval set to 10 minutes, 7 minutes, 5 minutes, 3 minutes, 1 minute, and 1 seconds. It is assumed that the road links that are candidates to match a divided point G_i are within 100m in radius from the point G_i, and nearest 5 road links are extracted. It means that the parameters are set as $threshold = 100$ and $J = 5$.

5.2 Evaluation Result

5.2.1 Overall Evaluation

To evaluate the map matching result, we compare road links M that is made by map matching to road links T that is a correct road links. The correct routes are made by human hand. It is defined that $N(A)$ means the number of road links that is composing road links A. By using these definition and formulas 5.1, 5.2, 5.3, $Same\ rate$, $Over\ rate$, and $Lack\ rate$ are obtained.

$$Same\ rate = \frac{N(M \cap T)}{N(M \cup T)} \tag{5.1}$$

$$Over\ rate = \frac{N(M \cap \overline{T})}{N(M \cup T)} \tag{5.2}$$

$$Lack\ rate = \frac{N(\overline{M} \cap T)}{N(M \cup T)} \tag{5.3}$$

The results are shown in Table 6. In every respect, the result of *simple real time method* is inferior to that of

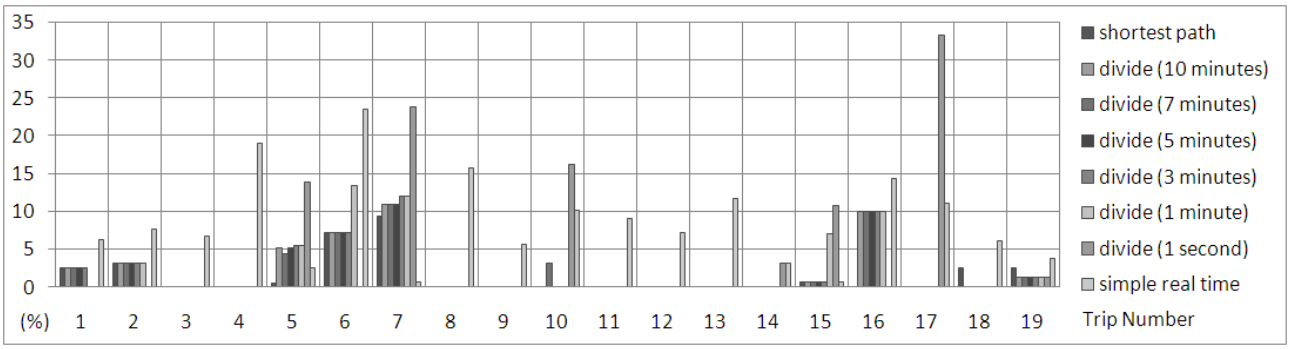

Figure 9. Over rate

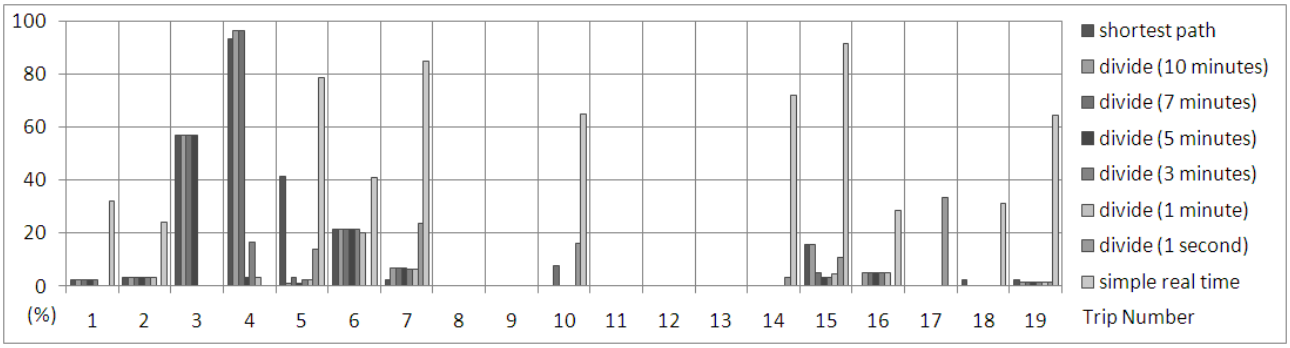

Figure 10. Lack rate

other methods. The proposed algorithms achieves about 90% $Samerate$.

$Same\ rate$ is highest with 5 minutes interval, and is lower with 1 minute and 1 second interval. Although $Lack\ rate$ rises as the interval becomes narrow, $Over\ rate$ falls. In addition, the matching result of 1 second interval shows that too narrow interval makes matching result worse.

It is also said that, $Same\ rate$ is lower in the case that division intervals are 10 minutes and 7 minutes because the detection error of circuit, turn, and roundabout course cannot be avoided in the shortest path method happens with wide interval.

5.2.2 Case Example

To understand the result more detailed, we investigate each matching result of 19 trips. $Over\ rate$ and $Lack\ rate$ are shown in Figure 9 and 10, respectively.

Running log of No.3 and No.5 is driving with the turn (Figure 6). Especially in No.5, $Lack\ rate$ has decreased remarkably because it was matched only to the road link around home in comparative methods. The running log of No.5 is a route that leaves home, takes someone to a station, and comes home. The proposed method improves $Lack\ rate$ by dividing.

Next, the matching result of No.8 is shown in Figure 11. This result is matched by division shortest path. It can be said that GPS points which has missed greatly from correct position are not able to match well suited road links even in the proposed method.

Moreover, in running log of No.19 whose results of every matching methods are good but not perfect, the constant match mistakes are generated regardless of division interval. Such errors occurred in the start point and the end point, which were also seen in other logs. Figure 12 is one example of this matching error. In this case, the system does not work correctly because the start point misses greatly from the correct road. In many cases, because driving beginning point is often a residential quarter, the electric wave of GPS cannot be correctly received. In addition, it is thought that the location information between them is unstable because the GPS receiver may require time to catch the GPS satellites.

5.3 Calculation Cost

We evaluate the calculation cost of our matching method to establish that it can run in real time. m is the number

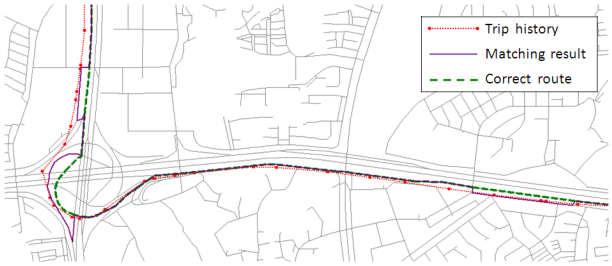

Figure 11. Matching result of No.8

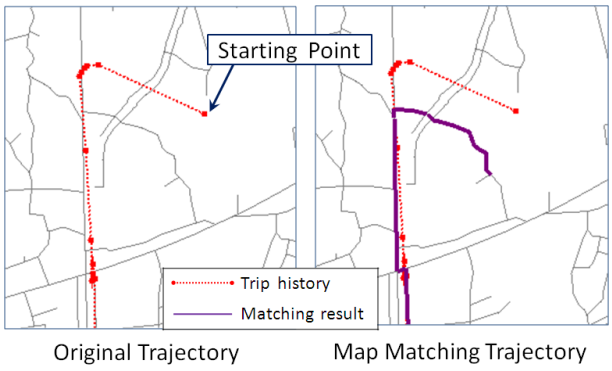

Figure 12. Matching error of the start point

of road links in the matching area, J is the number of road links in surroundings of selected GPS point, and g is the number of GPS points.

First, it is used $O(\log m)$ to look for the road links in surroundings of selected GPS point. Next, it involves $O(g \log m)$ to put a score on the road links in the map. Lastly, it is required $O(J^2 m \log m)$ to solve the shortest path problem J^2 times. Note that, J^2 is a constant value that we set $J = 5$ in the evaluation. Therefore, total calculation cost is summarized as $O((g + m) \log m)$. Our algorithm processes these calculation in every selected point, but because it divides the running trajectory into constant section, g and m are not so large. Therefore, the system can run our method can run in real time.

6 Conclusion

In this paper, we propose a map matching method for car navigation systems that predict user destination. Our method can make a correct route in real time. As a result of the evaluation, the advantage of our method to conventional methods is shown in most cases. When a running trajectory is divided at interval of 5 minutes, the result becomes the best.

As future work, it is necessary to improve the matching algorithm. Moreover, it is necessary to investigate the best division interval and scoring method.

Acknowledgments

This research was supported in part by Grantin-Aid for Scientific Research (A) (17200006) of the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

[1] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk: "On Map-Matching Vehicle Tracking Data," in *Proc. of the 31st Very Large Data Base Conference (VLDB)*, pp. 853–864 (2005).

[2] J. Greenfeld: "Matching GPS Observations to Locations on a Digital Map," in *Proc. of the 81th Annual Meeting of the Transportation Research Board* (Jan 2002).

[3] J. Yang, S. Kang, and K. Chon: "The map matching algorithm of GPS data with relatively long polling time intervals," in *Jour. of Eastern Asia Society for Transportation Studies(EASTS)*, pp. 2561–2573 (2005).

[4] V. Kostov, J. Ozawa, M. Yoshioka, and T. Kudoh: "Travel destination prediction using frequent crossing pattern from driving history," in *Proc. of IEEE Intelligent Transportation Systems*,Vol. 13, pp. 343–350 (Sep. 2005).

[5] T. Terada, M. Miyamae, Y. Kishino, K. Tanaka, S. Nishio, T. Nakagawa, and Y. Yamaguchi: "Design of a Car Navigation System that Predicts User Destination," in *Proc. of the 1st workshop on tools and applications on mobile contents (TAMC)*, pp. 54–59 (May 2006).