Making a Travel Diary from GPS Traces Using an Area-Based Reverse Geocoder

Kiichi Hikawa
Department of Computer Science and Engineering
Graduate School of Engineering
Nagoya Institute of Technology
Gokiso–cho, Syowa–ku,Nagoya–shi, Aichi, 466–8555 Japan
Email: hikawa@moss.elcom.nitech.ac.jp

Daisuke Yamamoto
Department of Computer Science and Engineering
Graduate School of Engineering
Nagoya Institute of Technology
Email: daisuke@nitech.ac.jp

Naohisa Takahashi
Department of Computer Science and Engineering
Graduate School of Engineering
Nagoya Institute of Technology
Email: naohisa@nitech.ac.jp

***Abstract*—This paper presents a system for creating a travel diary that shows a list of the locations visited on a trip with durations as well as a travel trace on a map. It also describes the implementation methods for the system. Proposed system determines the visited locations and infers the names of the sites and managed places with boundaries such as fences and roads, including the locations, from a GPS trace using an area-based reverse geocoder (ARG) newly developed for the system. Because it infers the names of sites using a site-boundary database, the ARG is more precise than conventional reverse geocoders, particularly when multiple sites are close to a portion of the travel trace. Proposed system introduces a sequence matching method that simultaneously matches consecutive GPS points with a boundary provided by the ARG. Because it uses site boundaries to find a sequence of noisy GPS points across the boundaries, the method effectively reduces the impact caused by GPS noise.**

***Keywords*—Web map, Reverse Geocoder, GPS, Travel Diary , Travel Trace, Area Database**

I. INTRODUCTION

Using embedded GPS (Global Positioning System) receivers, many mobile devices make it possible for users to record their locations and make travel tracing easy. A user can use GPS data with many GIS (Geographic Information System) applications, and share travel traces with other users. By storing GPS traces and sharing them with others, users can recall past travels, understand the travels of other people, and make new travel plans.

Several GIS applications draw a GPS trace as a line that follows the GPS points over a map view and display it on a screen. We have developed systems displaying travel traces [1]–[3]. These help users to easily see an overview of a trip. However, it can be difficult to grasp detailed information such as the name of a place visited (‘stayed at’) from the GPS trace on a map alone. In several studies, determining significant places from GPS traces has been addressed [4]–[6].

A GPS trace with a list of locations and visit (‘stayed’) durations for a trip will help users better understand their travel. The names of the sites, including the locations, are essential. Although conventional reverse geocoders provide the identification of a location, they provide only an address corresponding to the location and a list of names of sites close to that location. In other words, they cannot provide the name of a site including the specific location precisely.

Many WEB map services such as Google Map [7] provide the functions of a geocoder and reverse geocoder [8]. They can connect locations to sites, areas managed by someone and surrounded by boundaries like fences and roads, such as precincts of shrines, farms, and parks, which travelers may visit. They have pairs that include a site name and a representative point in their database. Given the name of a site, a geocoder will return the coordinates of the site-representative point, a latitude and longitude pair. On the other hand, given the coordinates of a point, a latitude and longitude pair, a reverse geocoder will return a list of candidate site names with the sites sorted by the distances between the point and the representative point of the sites. These functions are useful to find a site on a map. They have a problem, however, finding the name of a site precisely. They connect the name of a site to the coordinates of the site-representative point. Given the coordinates of a point inside a site, they may return the name of a different site from the site including the point, one that is close to the point.These problems occur when the point is inside a large site and close to its boundary when another small site is close to the point.

To resolve this problem, we have developed a site-boundary database and an area-based reverse geocoder, called ARG that uses this database. The site-boundary database connects a site to its name and boundary. Given the coordinates of a point, the ARG returns the name and boundary of the site that includes this point. In general, a significant effort is required to create a site-boundary database because it is more difficult to capture boundary data than representative point data for sites. To compensate for the lack of boundary data, we have developed a method for approximating this automatically as follows. The method calculates a loop road [9]–[11], a path surrounding the representative point of the site in the road network, and uses it as an approximate boundary of the site. If other site-boundaries overlap the loop road, it removes the overlapped areas from the loop road to improve the approximation.

The objective of our research is to precisely determine the sites visited on a trip with durations in each site using a GPS trace, and generate a travel diary list including a travel trace

 ISBN: 978-1-61208-320-9

on a map. The travel diary enables users to understand the trip intuitively and precisely.

The remainder of this paper is organized as follows.First, we describe the systems used to make travel diaries from GPS points in Section Ⅱ.We propose a system for making a precise travel diary using ARG in Section Ⅲ.In Sections Ⅳ, we describe the implementation methods for two key aspects of the proposed system, an area-based reverse geocoder and a sequence matching method, which matches a sequence of GPS points with the boundary of a site.We describe an experimental prototype of the proposed system in Section V and evaluate the proposed system by comparing the ARG and a conventional reverse geocoder through experiments determining locations and durations from GPS traces in Section VI.We conclude our presentation in Section Ⅶ.

II. Making a travel diary from GPS traces

A. Overview

We propose a system for generating a travel diary that analyzes GPS data that a traveler records during a trip, and shows the locations visited. It precisely determines the sites visited on the trip and durations spent in the sites from a GPS trace, and shows a travel diary listing with these sites as well as a travel trace on a map, as shown in Fig.1.

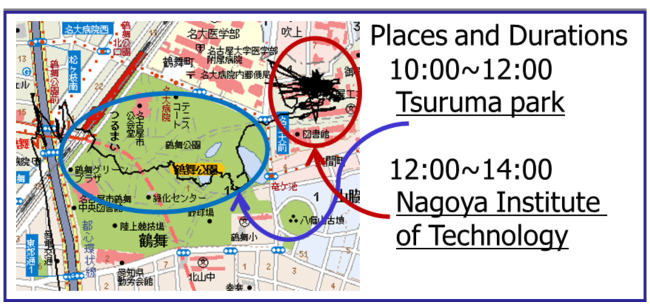

Fig. 1. Travel diary

B. Implementation Issues

In making a travel diary, extracting the portion of the GPS trace, that was recorded during the stay in the site and detecting the visited sites (‘stayed’) is important. Conventionally, reverse geocoders can be used for transforming a point in a GPS trace into the name of the location corresponding to the point.The duration that the traveler visited (‘stayed at’) the site is calculated from the sequence of consecutive GPS points corresponding to the site.

A GPS trace is a sequence of time-stamped points, for example, $P_i(i = 1, 2, ..., n)$, a 4-tuple of latitude, longitude, altitude, and time, that represents the geographic location, latitude, longitude, altitude, and the time the point was logged. Using a reverse geocoder, we can select the site shown by the reverse geocoder nearest to the GPS point of the representative point. This means that we can define a Voronoi diagram using the center coordinates (site representative points) of all the sites. We can then divide the area including the sites shown by the reverse geocoder into sub-areas determined according to the Voronoi diagram, and set boundaries by considering each sub-area as a site. Point matching classifies to which sub-area each point in the GPS trace belongs.

The above method may be unable to estimate the locations and durations of visits precisely for the following reasons. Consider the example of the GPS data recorded during the movement in the neighborhoods of sites A, B, and C as shown in Fig. 2. In this figure, X marks show the site-representative point and the dotted lines show the Voronoi diagram defined by the site-representative point. In this case, the sequence of the GPS points passes inside sites A and C. Although the sequence does not pass inside site B, point matching with a conventional reverse geocoder will record site B in a travel diary because it determines that the three GPS points in the middle of the sequence are inside the area of site B according to the Voronoi diagram as shown in a Fig. 2.

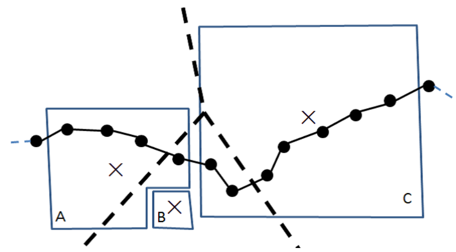

Fig. 2. Representative point of sites and Voronoi diagram

Figure 3 shows an example of a GPS trace that may include noise caused by large buildings or by narrow roads. It shows the movement between sites B and C. When moving frequently near the boundaries of the sites, as shown in this figure, point-matching will provide the location of a visit list showing that the traveler goes back and forth between sites for short intervals, such as site C→site A→site C→site A→site C. Erroneous position coordinates may be included in the GPS trace data because of noise. When a sequence fluctuates between sites, we can say that the sequence includes noise and the result of the point matching has a high possibility of being incorrect. This is because in many cases there are physical boundaries such as fences and roads between two adjacent sites and the traveler cannot cross these boundaries so frequently.

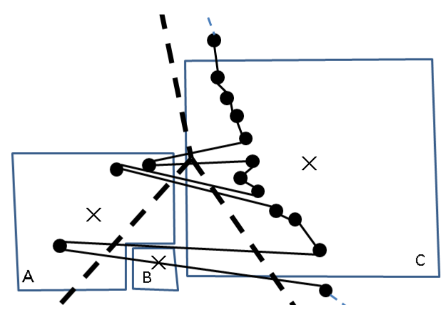

Fig. 3. GPS points which go to and return near a boundary

The requirements to implement a travel diary system are summarized as follows.

Requirement 1: It is required to solve the above-mentioned problem and to be able to estimate visited locations with high accuracy, even when two or more sites border.

Requirement 2: Because the results of matching a noisy sequence with site boundaries can be erroneous, it is necessary to implement a matching method that can reduce the impacts caused by GPS noise.

III. Proposed System

In order to satisfy the above requirements, we propose a travel diary system using GPS traces with the following features.

Feature 1 To satisfy Requirement 1, the system should estimate the location of visits and determine the name of the sites, managed places with boundaries such as fences and roads, including the locations from a GPS trace, using the area-based reverse geocoder (ARG) newly developed for this system. ARG is more effective than conventional reverse geocoders, particularly when multiple sites are close to some portion of the travel trace, because it determines the name of the sites more precisely using a site-boundary database.

Feature 2 To satisfy Requirement 2, the system should introduce a sequence matching method that simultaneously matches consecutive GPS points with a boundary provided by the ARG. The method must be able to reduce the impact caused by GPS noise, by using site boundaries to find a sequence of noisy GPS points across the boundaries.

The proposed system consists of a site-boundary database and the functions shown in Fig.4.

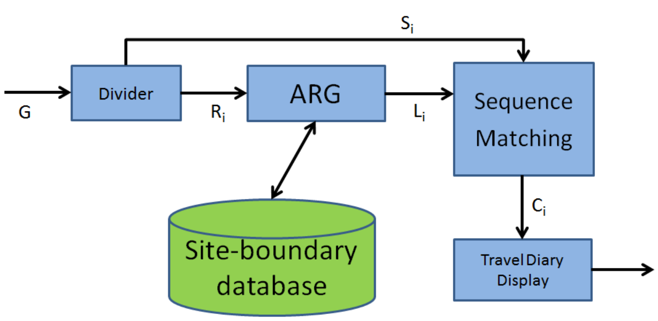

Fig. 4. Structure of the Proposed System

Divider: This function transforms a GPS log file, G, into a GPS trace, a sequence of points with five attributes, latitude, longitude, time, address, and name, where the first three attributes are set by copying the data of the corresponding point in the GPS log file and the last two attributes are empty. It also identifies an address for each point of the GPS trace using the ARG, and attaches it to that point as its attribute. Consecutive points having the same address attribute are gathered and formed into a sequence, S_i, where i=1,2,...,n and n is the number of sequences. The Divider calculates the minimal rectangle, R_i, which includes all the points of S_i for each i=1,2,...,n.

ARG: By searching a site-boundary database, this function obtains a list set of L_i, area data, where each entry is a pair of site name and boundary, for each set of all the sites in R_i for i=1,2,..,n.

Sequence Matching: This function matches the points in S_i with the boundaries stored in L_i and creates a chunk C_i of consecutive points with the three attributes, name, address, and duration. The name and the address attributes show the name and the address of the site that the traveler visited ('stayed at'). The duration attribute shows a time that the traveler stayed within the site.

Travel Diary Display: Using the chunks from the sequence matching, this function creates and displays a list of records, each a pair of the name of the site where the user visited and the time spent at the site.

Using these functions, the proposed system creates a travel diary from a GPS log file, G, by the following steps:

STEP1: The Divider function divides G into multiple sequences, $\{S_i\}$, and calculates a minimal rectangle R_i for each S_i in $\{S_i\}$.

STEP2: A list of the area data for all of the sites in R_i, L_i, is generated using ARG.

STEP3: Using S_i and L_i, the Sequence Matching builds a sequence of chunks, C_i.

STEP4: The Travel Diary Display function displays the attributes, a triplet of duration, site name, and address for each chunk in C_i.

IV. Implementation of the Proposed System

A. Implementation of Area-Based Reverse Geocoder

In order to create ARG, it is necessary to gather significantly more site area boundary data than is found in a typical site database. In a site database, many site area records include only a pair of name and representative point. Only a small number of site area records have boundary data as well. For the site records without boundary data, we build site boundaries automatically by calculating a loop road [9]–[11], a path surrounding the representative point of the site in the road network, and consider the boundary of the site area to be the loop road. If an area is surrounded by a loop road and used for multiple purposes, such as a school and park, we divide it into smaller areas using land usage data from a GIS database. These methods can generate boundary data for any site data consisting of a site name and its representative point.The result is that we can create a site-boundary database, a set of site data consisting of a site name and its boundary. We implement an area-based reverse geocoder (ARG) using this database and use it to generate our travel diary.

The site-boundary database has site data consisting of site name, boundary, representative point, and purpose.A boundary is shown by a polygon and the boundary data is a sequence of position coordinates of a polygonal vertex.

A1-A16 of Table I are classified according to the existence of each component of the site data. If data exists, it is indicated with the mark, ○, otherwise, it is marked, ×. Even when there is no data, it can be created easily from other data. For example, a representative point can be created from boundary data. This is expressed with the mark, △ .

In order for ARG to output a site name, a representative point, boundary, and purpose, such as sites A1-A4 of TableI, is required. Data such as A1 and A2 can be created by combining A5 and A6 that does not have boundary data, although it has a site name, with A9-A12 that has boundary data, but does not have a site name. That is, A5+A12 → A1 and A6+A12 → A2. In the case of overlapping boundaries where the use differs, a new boundary is generated by removing the overlapping portion.

TABLE I. SITE DATA

	Site name	Boundary	Representative Point	Purpose	Example of Data
A1	○	○	○	○	Tourist attractions (867 places in Japan)
A2	○	○	○	×	Address
A3	○	○	△	○	Named land usage data
A4	○	○	△	×	
A5	○	×	○	○	Public facility (9082 places in Aichi prefecture). Park (3938 places in Aichi prefecture)
A6	○	×	○	×	
A7	○	×	×	○	Shopping street, Busy street. Business district
A8	○	×	×	×	
A9	×	○	○	○	
A10	×	○	○	×	
A11	×	○	△	○	Unnamed land usage data
A12	×	○	△	×	Loop road
A13	×	×	○	○	
A14	×	×	○	×	
A15	×	×	×	○	
A16	×	×	×	×	

TABLE II. DEFINITION OF THE AREA DATA TABLE

column	type	description
id	integer	ID of the area data
name	text	site name of area data
lat	real	latitude of representative point
long	real	longitude of representative point
north	real	northernmost latitude of area data
south	real	southernmost latitude of area data
west	real	westernmost longitude of area data
east	real	easternmost longitude of area data
coordList	text	list of coordinates

The site-boundary database is indexed by the columns north, south, west, and east, in TableII. Given the coordinates of a point *p*, a pair of latitude *p*.lat and longitude *p*.long, the ARG returns an area $\bar{a}$ that includes this point from set A of all areas in the site-boundary database using the following STEPs.

STEP1: S1 = { *a*∈A | *a*.west < *p*.long ∧ *a*.east > *p*.long ∧ *a*.north < *p*.lat ∧ *a*.south > *p*.lat }

STEP2: Obtain the area data *a* ,which includes the point *p* , from the all elements in S1

B. Implementation of Sequence Matching

Point matching methods match each point of the GPS trace with a site boundary one at a time, and determine to which area they belong. Using ARG, methods simultaneously match a sequence of consecutive GPS points with a boundary provided by ARG and determine in which site each point of the sequence belongs. This is based on the hypothesis that many sites have boundaries that separate the inside and outside clearly, such as a fence and a road. When the sequence of consecutive GPS points cross the boundary provided by the ARG frequently, we consider that the crossing of the boundary is caused by the influence of GPS noise. When a sequence of GPS points crosses a road frequently as shown in Fig.5, the sequence matching method judges that it is an influence of noise. It determines that the minority points, points in site A in this figure, belong to the same site as the majority points, the points in site C in the figure, if the number of minority points is significantly smaller than the majority points.

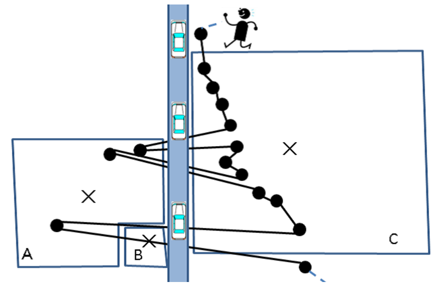

Fig. 5. GPS points that go and return near a boundary

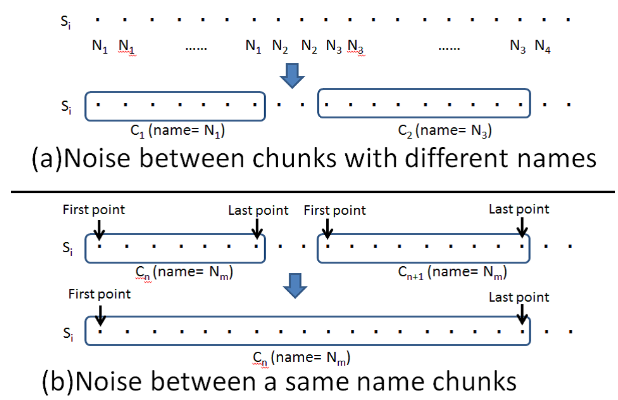

Fig. 6. Noise removed

In Fig.6 (a), given a sequence of points, S_i, the sequence matching function identifies a site name for each point of S_i, and attaches it to that point as the name attribute, N . It organizes the points of S_i by their name attribute. Consecutive points with the same attribute are gathered and formed into a chunk, C_i. The same name and address attributes as those of the gathered points is attached. When a chunk has only a few points, it is regarded as noise and discarded.In Fig.6 (b), when the name attribute of a chunk is the same as that of an adjacent chunk discarded in the previous step, these chunks are gathered and formed into a single chunk. The sequence matching calculates the duration stayed at the chunk for each chunk of C_i. Assume the time of the first point of C_i is t1, and the time of the last point of C_i is t2, the duration *d* is calculated by *d* = t2 - t1 . We introduce *d0* as the minimum duration of a stay in a chunk because we regard a short length chunk as noise. If *d* < *d0* then C_i is considered as noise. If *d* > *d0*, we set the value of the duration attribute of C_i to *d*.

V. PROTOTYPE SYSTEM

We implemented a prototype system of the proposed method using JAVA. The database used for the prototype system was MySQL. The following data was used as site data: address data（A2）and land usage data (A3, A11) from AlpsMAP [12], public facility data (A5) from the Ministry of Land, Infrastructure and Transport [13], and loop road (A12). The map picture used is from AlpsMAP.
The prototype system loaded the GPS trace data shown in Fig.7. It drew a GPS trace (①) as a line that followed the

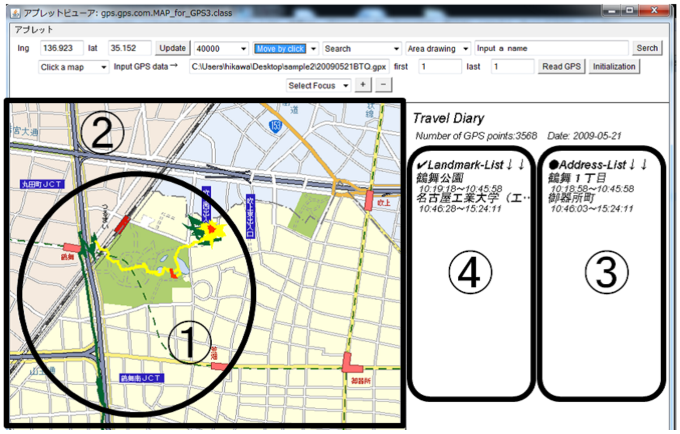

Fig. 7. Prototype System

GPS points over a map (②) on the left-hand side of the applet screen.
The list of locations and durations of each visit is shown on the right-hand side of the applet screen. First, the list of addresses is displayed on the extreme right-hand side (③). Then, the names of the sites are displayed, one by one, on the left-hand side (④) of the address name of the corresponding address. Nothing is displayed on the left-hand side of the address when a site is not found.

VI. Evaluation

We evaluated the accuracy of the locations and durations of each visit that the proposed system determined to generate the travel diary. We checked the list of the locations and durations of each visit that the prototype system created, and verified whether the site at which the traveler stayed was detected. This evaluated the usefulness of our ARG.

A. The Method of an Experiment

We incorporated the following two methods into the prototype system for evaluation purposes.

ARG-Method（Proposed method）: Using our ARG, it determines the location and duration of the visit.

RG-Method（Baseline）: Using a conventional reverse geocoder, it determines the location and duration of the visit. RG-Method simply selects the site shown by the reverse geocoder nearest to the GPS point of the representative point.

In both method, we set d0 in the same value, which introduced as the minimum duration of a stay. We used the thirty GPS traces that were recorded by moving inside Aichi.

First, we drew each GPS trace on a map, and showed this to the traveler. Next, in the GPS trace, the traveler reported the name of the site visited, and identified point P1 showing the entrance and point P2 showing the exit. We defined the reported site as the Correct Answer Site, that was actually visited. We defined the difference of the time between P2 and P1 as the duration Correct Answer Duration. When a traveler visited (‘ stayed at ’) in the same site more than once, we appended a numerical value showing the number of times stayed at the site name, and regarded it as a different site.

Next, using the ARG-Method (proposed method) and the RG-Method, we detected visits or stay sites, which we call Detect Sites. We determined the time stayed at these sites, which we call Detect Duration. When a traveler stayed in the same site more than once, we appended a numerical value to the site name, as we did with the Correct Answer Site and regarded it as a different site. Moreover, similar to the Correct Answer Duration, in the GPS trace, we determined P3, which was the first point and P4, which was the last point, in the Detect Site. We defined the difference of the time between P4 and P3 as the duration Detect Duration, that is, the time that the traveler stayed in the Detect Site.

Experiment 1: We obtained the following sets with each method from the GPS data.

V_1 : Set of Detect Sites

V_2 : Set of Correct Answer Sites

$V_3 = V_1 \cap V_2$

We obtained the number n_1,n_2,n_3 of elements of the sets V_1,V_2,V_3 and calculated P and R using the following formulas:

Precision (p_v) $= n_3/n_1$

Recall (r_v) $= n_3/n_2$

We calculated F_v using the following formula:

$F_v = 2 / (1/r_v + 1/p_v)$

Experiment 2: In the GPS trace, we evaluated the validity of the Detect Duration detected by the two methods by comparing the GPS points in Correct Answer Duration with the GPS points in Detect Duration as follows. We obtained the following sets that consisted of the chunks showing the duration for each site from each site contained in the set V_3 of Experiment 1 as follows:

T_1 : Set of GPS points in Detect Duration

T_2 : Set of GPS points in Correct Answer Duration

$T_3 = T_1 \cap T_2$

We obtained the number m_1,m_2,m_3 of elements of the sets T_1,T_2,T_3 and calculated P and R using the following formulas:

Precision (p_t) $= m_3/m_1$

Recall (r_t) $= m_3/m_2$

We calculated F_t using the following formula:

$F_t = 2 / (1/r_t + 1/p_t)$

B. Results

The results of Experiment 1 for the thirty GPS traces are shown in FIg.8, where the horizontal axises are GPS trace numbers and the vertical axises are the Precisions, Recalls and F values. It shows that the proposed ARG-Method detected the stay sites correctly with Precision=1 and Recall=1 for the GPS traces numbers 1, 2, 7, 8, 19, 24, 27, and 28. However, the Precision and Recall are low for the GPS trace numbers 4, 5, 9, 11, 13, 14, 15, 20, 22, 25 and 26. On the other hand, although the RG-Method detected all the stay sites (Recall=1) for the GPS traces numbers 2, 7, 9, 11, 12, and 19, the Precision is very low. We can say that the RG-Method provides low precision and the RG-Method detected a site that was not actually visited for many GPS traces. Comparing the F_v

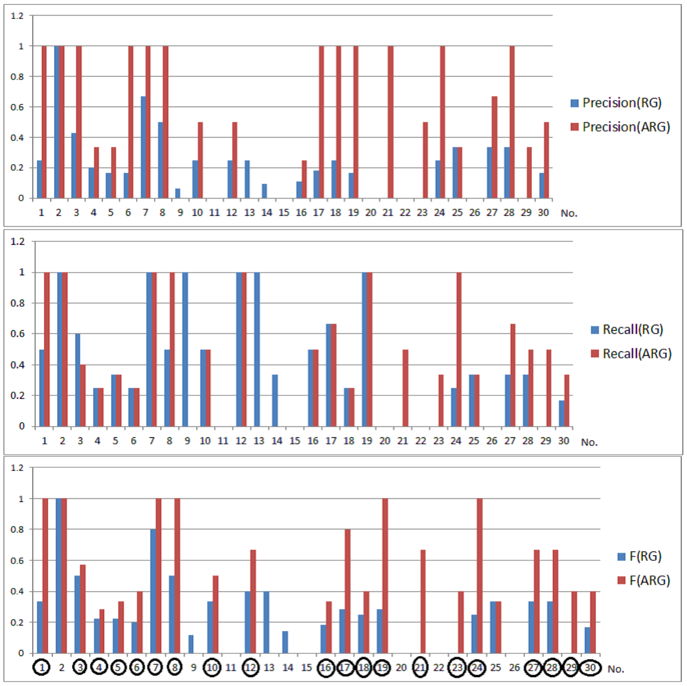

Fig. 8. Results of Experiment 1

values, the ARG-Method is better than the RG-Method for many GPS traces. Based on this, in this experiment, the results indicate that the stay site accuracy determined by the ARG-Method is higher than that of the RG-Method. However, the Precision, Recall, and F value were also low for the ARG-Method in some cases. This was caused by the following: (1) Because there was only a small amount of data in the database and there was no area data for a stay site, the system could not detect the site. (2) When two or more sites exist in one area, the system could not detect a stay site correctly. In other words, to improve the effectiveness of the site detection of the proposed method, it is necessary to expand the site data database. We present the results of Experiment 2 in Table III. As shown in Table III, the proposed ARG-Method recorded

TABLE III. RESULTS OF EXPERIMENT 2

No	p_v		r_v		F_v	
	RG-Method	ARG-Method	RG	ARG	RG	ARG
Average	0.883	0.951	0.779	0.857	0.828	0.902

a higher value for Precision, Recall and F value, compared to the RG-Method. This shows that when the ARG-Method identifies a site correctly, it can calculate the duration very accurately. The value of Recall for the RG-Method is low, and even when a stay site is detected correctly, there is a tendency to calculate a shorter duration than actual. By comparing the F_t values in Table III, we determined that the accuracy of the duration calculated by the ARG-Method is higher than that of the RG-Method. The above experiments with thirty GPS data showed that the proposed method is effective. However we must evaluate the proposed method with more GPS data in order to clarify its effectiveness.

VII. CONCLUSIONS

In this paper, we presented an area-based reverse geocoder (ARG) and described an implementation method. Moreover, we presented a system using ARG to create a travel diary and described the implementation method. Furthermore, we compared and evaluated the proposed system and a system using a conventional reverse geocoder, using locations and visit durations of a system-generated travel diary. As a result, we were able to derive the sites and durations at which the user stayed from the GPS trace with high precision using the ARG-Method, better than when the conventional reverse geocoder was used. There were. however, cases where a site was undetectable. To resolve this, we must expand the site data, which includes the area data. The subject of future research follows. When the representative point of two or more sites exist in the boundary of an area, it is difficult for ARG to show the exact position of the site. When detecting the site as ‘stayed’, a device using a combination of ARG and reverse geocoding by distance of points, is necessary.

ACKNOWLEDGMENTS

We would like to thank Yahoo! Japan Corporation for supporting us in the development of the prototype system. This work was also supported by JSPS KAKENHI 23500084 and 25700009.

REFERENCES

[1] Pablo Martinez Lerin, Daisuke Yamamoto, and Naohisa Takahashi “Making a Pictorial and Verbal Travel Trace from a GPS Trace”, Proceedings of the 11th International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2012), LNCS 7236, Springer, pp. 98-115 , 2012.

[2] Pablo Martinez Lerin, Daisuke Yamamoto, and Naohisa Takahashi, “Inferring and Focusing Areas of Interest from GPS traces”, Proceedings of the 10th International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2011), LNCS 6574, Springer, pp. 176-187 , 2011.

[3] Pablo Martinez Lerin, Daisuke Yamamoto, and Naohisa Takahashi, “Pace-Based Clustering of GPS Data for Inferring Visit Locations and Durations on a Trip,” IEICE Transactions on Information and Systems, 2014 to be published.

[4] YU ZHENG and XING XIE , “Learning Travel Recommendations from User-Generated GPS Traces”, ACM Transaction on Intelligent Systems and Technology, Vol. 2, No. 1, Article 2, 2011.

[5] Changqing Zhou, Dan Frankowski, Pamela Ludford, Shashi Shekhar, and Loren Terveen, “Discovering personally meaningful places: An interactive clustering approach”, ACM Transaction on Intelligent Systems and Technology, Vol. 25, No. 3, Article 12, 2007.

[6] Nobuharu Kami, Nobuyuki Enomoto, Teruyuki Baba, and Takashi Yoshikawa, “Algorithm for Detecting Significant Locations from Raw GPS Data”, Discovery Science Lecture Notes in Computer Science Vol. 6332, pp 221-235, 2010.

[7] GoogleMaps, January 2014. http://maps.google.co.jp/

[8] Google Maps API Web Services, January 2014. https://developers.google.com/maps/documentation/geocoding/

[9] Hiroki Ito,Daisuke Yamamoto, and Naohisa Takahashi, “Creating a database of roads around the object and method of fast derivation of the roads around the object by using database”, DEIM Forum 2011, B4-3, 2011.

[10] Hiroki Ito, Daisuke Yamamoto, and Naohisa Takahashi, “A Loop-Road Database Building Method Supporting Multiple Meshes” ,The 74th National Convention of IPSJ ,1P-6, 2012.

[11] Daisuke Yamamoto, Hiroki Itoh, and Naohisa Takahashi, “One Click Focusing: An SQL-based Fast Loop Road Extraction Method for Mobile Map Service”, Proceedings of the 4th International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2012), pp. 7-16, Valencia, Spain, 2012.

[12] AlpsMAP, November 2013. http://alpsmap.yahoo.co.jp/

[13] National Land Numerical Information Download Service, November 2013. http://nlftp.mlit.go.jp/ksj/

