PAPER *Special Section on Data Engineering and Information Management*

Pace-Based Clustering of GPS Data for Inferring Visit Locations and Durations on a Trip

Pablo MARTINEZ LERIN[†*a)], **Daisuke YAMAMOTO**[†,††b)], ***Nonmembers, and*** **Naohisa TAKAHASHI**[†c)], ***Member***

SUMMARY Travel recommendation and travel diary generation applications can benefit significantly from methods that infer the durations and locations of visits from travelers' GPS data. However, conventional inference methods, which cluster GPS points on the basis of their spatial distance, are not suited to inferring visit durations. This paper presents a pace-based clustering method to infer visit locations and durations. The method contributes two novel techniques: (1) It clusters GPS points logged during visits by considering the speed and applying a probabilistic density function for each trip. Consequently, it avoids clustering GPS points that are near but unrelated to visits. (2) It also includes additional GPS points in the clusters by considering their temporal sequence. As a result, it is able to complete the clusters with GPS points that are far from the visits but are logged during the visits, caused, for example, by GPS noise indoors. The results of an experimental evaluation comparing our proposed method with three published inference methods indicate that our proposed method infers the duration of a visit with an average error rate of 8.7%, notably outperforming the other methods.

key words: *clustering method, visit inference, place inference, duration inference, GIS*

1. Introduction

Mining knowledge from the large amount of travelers' GPS traces already available facilitates new travel recommendation and travel diary generation applications [1]–[3]. Some of the most valuable pieces of information inferred from GPS traces are from trip visits, i.e., temporary stays at locations doing activities that may be relevant for remembrance and recommendation.

Many existing works have proposed methods to infer visits, known as place-inference methods, and have evaluated the methods' abilities to infer the location of each visit [4]–[9]. However, to the best of our knowledge, no study has been conducted to evaluate their ability to infer the duration of each visit. We argue that inferring the duration of each visit (i.e., the time spent during each visit) is essential to answering travel recommendation and travel diary generation queries. Here are several example applications:

- Recommendation of visits, and the trace followed during the visits, that take a determinate amount of time;
- Provision of the amount of time that travelers often spend during a visit to a determinate location; and
- Generation of a detailed description of a past trip.

Among the existing works, clustering GPS points from a trip's GPS trace has been proven a useful and popular method for inferring visit locations [4]–[7]. Inferring the visit locations does not necessarily require the clusters to be precise and complete, i.e., to include all and only the points logged during each visit, because locations are usually inferred using the center of each cluster. Inferring the visit durations, on the other hand, requires the clusters to be precise and complete because durations are inferred using all the points within each cluster. However, making precise and complete clusters that represent both indoor and outdoor visits has the following two main challenges for current place-inference methods.

On the one hand, generating precise clusters, i.e., preventing the clustering of GPS points not logged during visits but logged close to visits, especially with visits to locations of different shapes (e.g., elongated, rectangular, circular) and different sizes (e.g., a downtown region and a museum, is a challenge. This is because place-inference methods usually cluster together all points within a radius (circular shape), that are predefined and the same for all trips, in order to infer locations where travelers have spent a relatively long time.

On the other hand, generating complete clusters, i.e., clustering GPS points logged during visits but dispersed (e.g., when travelers move quickly during outdoor visits) and far from the visits (e.g., when there is high GPS noise during indoor visits), is also a challenge. This is because place-inference methods usually consider the GPS points as independent points in space.

These challenges generate two questions that, to the best of our knowledge, have not been addressed in existing works: (1) How good are current place-inference methods when inferring the visit durations? (2) What techniques can be applied to handle the challenges?

To answer these questions, we propose a pace-based clustering method designed to infer the duration and location of indoor and outdoor visits, and present a study conducted on duration and location inference that evaluates our proposed method and compares it to three published place-inference methods. Our proposed clustering method addresses the challenges affecting the inferring of visit durations by applying the following two key ideas.

Manuscript received July 4, 2013.
Manuscript revised October 29, 2013.

[†] The authors are with the Graduate School of Engineering, Nagoya Institute of Technology, Nagoya-shi, 466–8555 Japan.
[††] The author is also with CREST, JST, Tokyo, 102–0076 Japan.
[*] Presently, with Hitachi Ltd., Tokyo, Japan.
a) E-mail: pablo@moss.elcom.nitech.ac.jp
b) E-mail: yamamoto.daisuke@nitech.ac.jp
c) E-mail: naohisa@nitech.ac.jp

DOI: 10.1587/transinf.E97.D.663

First, the proposed method clusters GPS points using a speed threshold in a fragmentation step. Since the distance between points is not considered, this avoids clustering points that are close to but unrelated to visits, and is able to generate clusters of different sizes and shapes. In addition, a suitable speed threshold is computed for each GPS trace, by using a probability density function (pdf) of the traveler's speed, to adjust the threshold to different kinds of trips, such as trips done mainly by walk and done mainly by car. As a result, the method can generate clusters that are more precise.

Next, in a defragmentation step, the proposed method merges clusters that belong to the same visit, and then completes the clusters using additional GPS points that were logged during visits but not clustered, by considering their temporal sequence, to cluster points that are logged during visits but dispersed or far from the visit. As a result, the method can generate clusters that are more complete.

The concepts underlying our proposed method and experimental results that confirm its feasibility have already been presented in our previous work [10]. The most important difference with this work is that while our previous work discussed only the location inference, this work discusses location and duration inference. The most important feature of this work is its precise duration inference. More specifically, this work is different in two main aspects.

The defragmentation step proposed in this work includes a new rule (referred to as the second rule) that enhances the inference of the durations of visits by dealing with high GPS noise indoors, which was not one of the objectives of our previous work. In fact, the method proposed in our previous work focuses on outdoor areas and subareas of interest, while the method proposed in this work focuses on indoor and outdoor visits.

The study presented in this work is different in that it evaluates the duration inference, compares our proposed method with existing methods, and identifies our method's shortcomings, which our previous work does not. Further, the amount of GPS data evaluated in this work is twice that used in the previous work.

The remainder of this paper is organized as follows. Section 2 describes related work on place inference. Section 3 defines the concept of a visit and discusses the requirements of place-inference methods. Section 4 describes our proposed pace-based clustering method. Section 5 explains the experimental evaluation and presents its results. Section 6 discusses the results and the main shortcomings of the evaluated methods. Finally, conclusions and directions for future work are given in Sect. 7.

2. Related Work

In this section, we describe place-inference methods that use coordinate-based systems such as GPS as source data. Some place-inference methods use fingerprints (e.g., Wi-Fi and GSM radio) as source data [11]. However, records from travelers are more popular and available as GPS data than as fingerprint data.

Wolf et al. conducted a study of a method to automatically generate a travel diary [3]. Their study used GPS loggers installed on the participants' cars and inferred a visit when the logged car speed was zero or near-zero for more than a short time period. A disadvantage of this method is that GPS data from a mobile device carried while traveling cannot be used.

Several works analyze GPS signal loss to detect visits to indoor locations. Marmasse et al. proposed a place-inference method that uses signal loss and distance between successive GPS points to identify candidate visits [12]. Candidate visits are considered meaningful based on their frequency. Ashbrook and Starner proposed a method that predicts users' movements using a Markov model [4]. Their method infers candidate visits at locations where the GPS signal is lost for more than a certain time period and then merges the candidate visits in a clustering step. A disadvantage of these methods is that they are unable to infer visits to outdoor locations.

Zheng et al. proposed a travel recommendation system that implements a radius-based clustering method that searches stay points [2], [13]. A stay point is a geographical region where a traveler stayed for more than a certain period of time within a particular distance. Although this method can only generate clusters of a predefined radius, it can infer both indoor and outdoor visits.

Kang et al. proposed a time-based clustering method to infer the visits during a trip [5]. The method clusters all GPS points within a particular radius and then prunes the clusters using a particular time period to retain only the meaningful visits. A step is used to merge near clusters. The novelty of this method is that Kang et al. use a temporary buffer to reduce the effects of GPS noise. By using a buffer, a new cluster is closed only when a particular number of successive GPS points are outside the cluster.

Several works have studied a density-based clustering method that implements the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [7], [14] to infer the visits during a trip. DBSCAN is a well-known algorithm that can make clusters of arbitrary shapes [15]. The method uses thresholds for the distance (proximity) and number of points (density) in order to make clusters and allow them to expand.

The place-inference method presented in this paper differs from existing works in two main ways.

First, our method clusters GPS points by analyzing their speed, while existing works analyze their position or density. Moreover, our method uses a threshold speed computed using the pdf based on each GPS trace, while existing works do not adjust the thresholds on the basis of each trip but use the same thresholds for every input trip.

Second, our method merges clusters that belong to the same visit, and completes the clusters with additional GPS points logged during the visits but unclustered by considering their temporal sequence. In contrast, existing methods do not consider the temporal sequence of the points when merging clusters and do not include additional points; they simply join the points within the clusters. Further, our proposed work merges clusters in order to avoid bad inference in the duration caused by high GPS noise indoors. This is in contrast to, for example, work done by Kang et al. [5]], which merges clusters in order to join several clusters of GPS points logged at very different time periods in the same location, because their work considers all the different time periods as the same visit.

Place-inference methods that use probabilistic clustering (e.g. [8], [9]) are outside the scope of our research.

3. Inferring Visits During a Trip

In this section, we define relevant concepts used in the rest of the paper. First, we define a visit and its attributes. We then discuss the input and output as well as the requirements of methods that infer visits during a trip.

3.1 Trip Visits

This work focuses on inferring visits during a trip, in particular, their location and duration attributes, which are defined as follows:

Visit: A visit $\upsilon_{\mathrm{T,P}}$ occurs when a traveler realizes meaningful activities (e.g., having lunch with a friend), i.e., activities that may be relevant for remembering and/or recommending, during one or more time periods on a trip T at location P (e.g., a restaurant). Roughly, we consider visits as the places a traveler would state when asked, “Where have you visited?” after a trip. Let us imagine two trips. TripA: The traveler went by car to a big downtown region, and walked around for three hours, doing quick sightseeing and window shopping without stopping. We consider that TripA has one visit to a downtown region for three hours. TripB: The traveler went by car to a big downtown region, and walked around, stopping at a museum for one hour and at a shop for 40 minutes. We consider that TripB has two visits, a museum for one hour and a shop for 40 minutes.

Location of a Visit: The location of a visit $\upsilon_{\mathrm{T,P}}$ is the geographic region that contains location P, usually delimited by a perimeter (e.g., a building, a park).

Duration of a Visit: The duration of a visit $\upsilon_{\mathrm{T,P}}$ is the total time spent at location P on trip T doing meaningful activities. When a visit to a location occurs during different time periods, the duration is the sum of all the time periods. We consider the time spent doing an activity at a location as the time difference between the moment the traveler enters and the moment the traveler exits the perimeter.

3.2 Inference Methods

The place-inference methods considered in this research receive a GPS trace as input and return a sequence of GPS point clusters as output. We define GPS point, GPS trace, and cluster, and then describe the criteria we use to infer visits from clusters below.

GPS Point: The GPS data obtained by a GPS logger usually represents a GPS point as a 4-tuple of (latitude, longitude, altitude, time). In this work, we define a GPS point p as a 4-tuple of ($p.latitude$, $p.longitude$, $p.speed$, $p.time$) that represents the geographic location (*latitude* and *longitude*) and *speed* of the traveler and the *time* the point is logged. We estimate the speed of a GPS point by dividing the distance by the time difference from the previous GPS point logged. The estimation has an error because the path length between both points is approximated as the direct distance between them without considering altitude, and the speed is an average of the speeds used between both points. However, the error is small because the GPS points are logged using a very short time interval (e.g., 5 s). In particular, these estimation errors are negligible when we move slowly. This is important because the accuracy of place inference depends on travelers’ behaviors in slow-speed movements.

GPS Trace: A GPS trace represents the travel trace of a trip. A GPS trace T is a sequence of n GPS points $p_{\mathrm{i}}(i = 1, \ldots, n)$ sorted temporally as follows: $T = [p_1, p_2, \ldots, p_{\mathrm{n}}]$, where $p_{\mathrm{i}}.time < p_{\mathrm{i+1}}.time \ \forall 1 \leq i < n$.

Cluster: A cluster of a GPS trace consists of a set of GPS points from the GPS trace. The points in the cluster may not follow the same order and may not be adjacent within the GPS trace.

From Cluster to Visit: Each cluster returned by a place-inference method represents a different visit during a trip. However, when several clusters infer the same location (as defined below), we consider that those clusters represent the same visit.

From Cluster to Visit Location: A visit location is inferred from a cluster as the geographic region at the cluster’s center (e.g., by reverse geocoding or visual examination over map images). The cluster’s center is defined as the average location of its GPS points.

From Cluster to Visit Duration: The duration of a visit is inferred from a cluster as the sum of the time difference between each pair of GPS points in the cluster that are adjacent within the input GPS trace. When a visit is represented by several clusters, the total duration is the sum of the durations inferred from each cluster.

Some existing works propose inferring the duration of a visit using the time difference between the first and last GPS points in the cluster [5]. We argue that using only the first and last GPS points may result in an inferred duration that differs significantly from the actual duration in two cases: (1) when a traveler visits a location on a trip during several time periods and (2) when a method clusters GPS points that are close but unrelated to a visit.

3.3 Requirements

To infer the visits of a trip, we define two requirements for place-inference methods, considering the two challenges mentioned in the Introduction.

Requirement R1: Generate a precise cluster, i.e., a cluster with just the points logged during a visit, even if the points form an arbitrary shape in the space.

Requirement R2: Generate a complete cluster, i.e., one with all the points logged during a visit, even if several of the points are far or dispersed in the space.

Methods that fail to fulfill requirement R1 tend to cluster GPS points logged during a visit with unrelated points. This may cause incorrect inference of the location, and the inferred duration may be longer than the actual duration. Methods that fail to fulfill requirement R2 tend to miss dispersed or far GPS points logged during a visit. This may result in the inferred duration being shorter than the actual duration.

4. Pace-Based Clustering

This section describes our proposed method, a pace-based clustering method to infer visit locations and durations from their GPS trace.

4.1 Overview

This method consists of three steps: (1) A fragmentation step in which GPS points logged during visits from the input GPS trace are clustered. (2) A defragmentation step in which clusters are merged and completed with GPS points logged during the same time period of a visit. (3) A visit extraction step in which spurious clusters are discarded. The three steps are described in detail in the following subsections.

Fundamentally, the method applies a probabilistic density function (pdf) of the traveler's speed to find the traveling paces used on a trip, and then clusters the GPS points logged when travelers move at the slowest pace for more than a considerable amount of time (e.g., 10 min). Considering the two example trips described in the previous section; in TripA, the traveler uses two traveling paces: car and walk. Because walking is the slowest pace, the proposed method clusters the sections of GPS trace where the traveler was walking for more than a particular amount of time, i.e., the section of the downtown region. In TripB, the traveler uses three traveling paces: car, walk, and almost stopped. Because almost stopped is the slowest pace, the proposed method clusters the sections of GPS trace where the traveler almost stopped for more than a certain amount of time, i.e., the sections of the museum and the shop.

The key ideas underlying our proposed method and its requirements are described below. They are motivated by the two requirements defined in the previous section and the following four observations, which were described in detail in our previous work [10]: (1) Travelers usually move at a slower pace during visits. (2) Speed thresholds may be used to distinguish to some extent the different paces during a trip (e.g., a pace is traveled at a speed below 5 km/h or a speed between 5 km/h and 15 km/h). (3) The number of paces and the speed thresholds that distinguish them often varies for different travelers and trips (e.g., walking versus driving trips). (4) When some of the points logged during a visit are dispersed in the space, some clustering methods tend to cluster the points that are not dispersed using more than one cluster.

Key Idea 1: Considering requirement R1 and observations 1 and 2, the proposed method clusters the input GPS points based on their speed (using a speed threshold). It enables the generation of more precise clusters (R1) than existing methods because existing methods cluster all the points within a radius (distance threshold).

Key Idea 2: Considering requirement R1 and observation 3, the proposed method uses a pdf that analyzes all the GPS points' speeds in order to automatically identify a suitable speed threshold for a particular traveler and trip. It enables the generation of more precise clusters (R1) than existing methods because existing methods use the same fixed threshold for all travelers and trips.

Key Idea 3: Considering requirement R2 and observation 4, the proposed method merges clusters that contain GPS points logged during the same visit, and completes them with additional GPS points that were not in the clusters but were logged during visits, by considering the sequence of the GPS points in the GPS trace. It enables the generation of more complete clusters (R2) than existing methods because existing methods merge clusters by simply joining the points within the clusters.

The pdf imposes a requirement on the approach: it requires that a relatively high number of GPS points represent the slowest traveling pace. Therefore, two considerations must be noted: (1) Input GPS trace should be logged in a time interval, rather than a distance interval. In cases where the GPS trace is logged in a distance interval, a preprocessing step should interpolate the GPS points in a time interval. (2) An input trip should be of a relatively long duration (e.g., more than 20 min) to distinguish the traveling paces. We believe trips are usually longer than 30 min. However, in case of very short trips, a default speed threshold for the common walking pace can be used (e.g., 7 km/h).

4.2 Fragmentation

The fragmentation step clusters the GPS points from the input GPS trace. This step consists of the two procedures described in detail in the following subsections.

4.2.1 Slowest Pace Retrieval

Given the input GPS trace T, this procedure retrieves a speed threshold $sThr$ that distinguishes the slowest pace used in T, following the key idea 2. First, the procedure builds the pdf $f'(s)$ of the traveler's speed s from T. It then returns the first local minimum of $f'(s)$ as the speed threshold $sThr$. The function $f'(s)$ and its first local minimum are defined below.

We define a cumulative distribution function (cdf) $F(s)$ and a pdf $f(s)$ of the traveler's speed s and a GPS trace T as follows. In our experiments, we use a value $\triangle s = 0.5$ km/h.

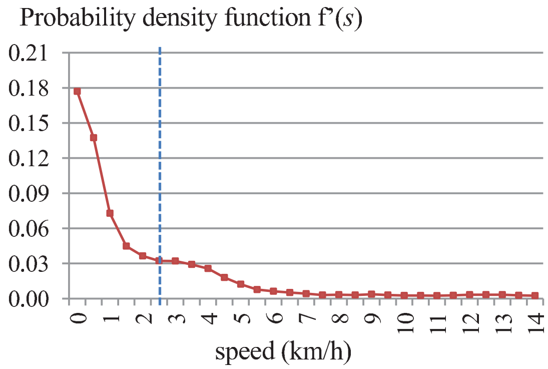

Fig. 1 Probability density function $f'(s)$ of the GPS trace T1 (see Table 1 in Sect. 5), which primarily involved car travel.

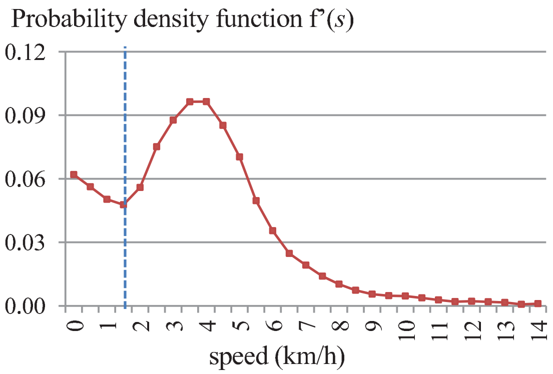

Fig. 2 Probability density function $f'(s)$ of the GPS trace T22 (see Table 1 in Sect. 5), which primarily involved walking.

$$F(s) = \frac{|\{p|p.speed{<}s,\ p{\in}T\}|}{|T|} \tag{1}$$

$$f(s) = \frac{d(F(s))}{ds} = \frac{|\{p|s{-}\triangle s{<}p.speed{\le}s,\ p{\in}T\}|}{|T|} \tag{2}$$

We consider that when a considerable number of GPS points in a GPS trace have similar speeds they represent a traveling pace used in a trip. Therefore, the local minimums in the pdf are inferred as speed thresholds that distinguish the traveling paces used in a trip. However, depending on the GPS data and the $\triangle s$ value, there may appear small local minimums that do not distinguish a traveling pace. Therefore, we define a new function $f'(s)$ that smoothes function $f(s)$.

$$f'(s) = \frac{f(s{-}\triangle s) + f(s) + f(s{+}\triangle s)}{3} \tag{3}$$

Assuming that the input GPS trace represents a trip over a considerable amount of time, any anomalous GPS speed of a few points caused by GPS error does not significantly affect the pdf because those points are a very small fraction of the total number of GPS points. Figures 1 and 2 show plots of function $f'(s)$ for the GPS traces T1 and T22, respectively, described in Table 1 in Sect. 5. The figures show $f'(s)$ for $0 \le s \le 14$ km/h and $\triangle s = 0.5$ km/h. For $s > 14$ km/h, the values of $f'(s)$ maintain the same tendency and are not relevant.

The first local minimum in $f'(s)$ distinguishes the slowest pace used in a trip. It may be detected manually by examining the plot of $f'(s)$, although it is a subjective process. We define the first local minimum as follows, in order to find it automatically.

First local minimum: The minimum speed s that fulfills $f'(s{-}\triangle s) > f'(s) < (\varphi + f'(s{+}\triangle s))$.

The dashed lines in Figs. 1 and 2 indicate the first local minimum found automatically in the represented functions. We use a small margin, $\varphi = 0.0005$, to compare $f'(s)$ and $f'(s{+}\triangle s)$ because they may have a similar value, as shown in Fig. 1. In all the GPS traces evaluated, we observed that the first local minimum is not related to the average speed of the GPS trace. Indeed, in the GPS traces in Figs. 1 and 2, the average speeds were 12 km/h and 3 km/h, respectively.

In this work, we use only the first local minimum in $f'(s)$. Searching for more local minimums in $f'(s)$ allows us to identify more traveling paces used in the trip. In our previous work, two local minimums in $f'(s)$ are used to infer and highlight on a map screen the areas and subareas of interest of a travel trace [10].

4.2.2 Trace Sections Retrieval

Given the speed threshold *sThr*, retrieved in the first procedure, and the input GPS trace T, this procedure returns a set of clusters that represent sections of T, following the key idea 1.

The procedure groups all the points in T with a $speed \le sThr$ into a set of clusters such that each cluster C fulfills the two following conditions.

Condition C1: The cluster's points are adjacent and follow the same order as in T, i.e. $T = [p_1, p_2, p_3, \ldots, p_n]$, $C = [p_x, p_{x+1}, \ldots, p_{x+m}]$, $x \ge 1$, $x{+}m \le n$.

Condition C2: The minimum number of clusters are used, i.e. $C = [p_x, p_{x+1}, \ldots, p_{x+m}]$, $p_{x-1}.speed > sThr$, $p_{x+m+1}.speed > sThr$, $p_i.speed \le sThr\ \forall x \le i \le x{+}m$.

4.3 Defragmentation

The defragmentation step merges and completes the set of clusters *SetC* retrieved in the fragmentation step, following key idea 3. This step consists of the following three points that use the two rules defined below: (1) Sort the clusters temporally in *SetC* in the order they are traveled. (2) Apply the first rule to all clusters. (3) Apply the second rule to all clusters.

First Rule: Two clusters $C_X = [p_A, \ldots, p_B]$ and $C_Y = [p_E, \ldots, p_F]$ that are adjacent within *SetC* are merged into one cluster $C_{XY} = [p_A, \ldots, p_F]$ when the sequence of points $S = [p_{B+1}, \ldots, p_{E-1}]$ between C_X and C_Y has an extension smaller than a threshold *extThr*. The extension of a sequence of points is computed as the sum of the Euclidean distance between each pair of adjacent points.

The first rule avoids bad inference by completing the clusters with dispersed points logged during visits, caused by the traveler walking at an irregular pace during a small fraction of a visit to a large outdoor location (e.g., a big park).

Second Rule: Two clusters $C_{\mathrm{X}} = [p_{\mathrm{A}}, \ldots, p_{\mathrm{B}}]$ and $C_{\mathrm{Y}} = [p_{\mathrm{E}}, \ldots, p_{\mathrm{F}}]$ that are adjacent within *SetC* are merged into one cluster $C_{\mathrm{XY}} = [p_{\mathrm{A}}, \ldots, p_{\mathrm{F}}]$ when the centers of each cluster are closer than a distance threshold *distThr*. The center of a cluster is computed as the average location of its GPS points.

The second rule avoids bad inference by completing the clusters with far points logged during visits, caused by high GPS noise during visits to indoor locations.

4.4 Visit Extraction

This step computes the duration of the clusters returned by the defragmentation step and discards the clusters with duration shorter than a small time threshold, *timeThr*, e.g., 10 min; these are considered spurious clusters. The remaining clusters are the output of the method. Each remaining cluster represents a visit, except when a visit occurs during different time periods, in which case each time period is represented by a different cluster. A time threshold is used because the time spent in a location seems to be a strong indicator of the importance of a visit [16]. This step is also used in related work [4], [5].

Table 1 Dataset used in the evaluation.

GPS trace	Number of GPS points	Trip distance (km)	Trip duration (hours)	Number of visits	Main features		
T1	2387	40	5.8	5	urban	car	outdoor
T2	3450	93	7.0	6	urban	car	indoor
T3	1185	26	2.6	2	urban	walk	outdoor
T4	3530	62	6.1	5	suburban	car	indoor
T5	2704	15	4.0	5	suburban	car	outdoor
T6	4277	11	6.3	4	suburban	walk	outdoor
T7	3809	51	7.2	5	urban	car	indoor
T8	3618	107	5.8	3	suburban	car	outdoor
T9	3742	84	6.4	5	suburban	car	indoor
T10	1909	16	3.0	3	urban	car	indoor
T11	4044	118	5.8	2	suburban	car	outdoor
T12	1128	20	1.6	2	suburban	car	indoor
T13	3352	74	5.5	5	suburban	car	outdoor
T14	3318	62	7.0	3	suburban	car	outdoor
T15	3682	56	5.3	4	urban	car	outdoor
T16	4392	60	6.3	6	suburban	car	indoor
T17	3303	116	5.4	3	suburban	car	indoor
T18	5744	133	8.2	6	suburban	car	indoor
T19	2824	28	5.9	5	urban	walk	outdoor
T20	5849	177	9.2	8	urban	walk	indoor
T21	4887	76	6.9	3	suburban	walk	indoor
T22	833	4	1.2	2	urban	walk	outdoor
AVG	3362	65	5.6	4.2			
SUM	73967	1429	122	92			

5. Evaluation

This section compares our proposed method with three existing methods via an experimental evaluation.

5.1 Experiment Setup

The data used for the evaluation was logged in 22 trips made on different days over a span of two years. Nine trips occurred within urban areas, while the remaining 13 trips spanned urban and suburb areas. Regarding the transportation mode, 16 trips were primarily by car, while the remaining six trips included a considerable amount of walking combined with the car. The trips were made for sightseeing purposes. In particular, half of the trips comprised mainly indoor sightseeing, i.e., visits to buildings such as museums, and the other half of the trips comprised mainly outdoor sightseeing, i.e., visits to wide open areas such as parks. The information for each trip is shown in Table 1.

All the trips were recorded with a GPS logger (Qstarz GPS Travel Recorder BT-Q1300) using a sampling rate of 5 s/point.

We determined the locations and durations of the visits on each trip by visualizing each GPS trace on detailed map and satellite images. The visualization included markers pointing to the center of the clusters inferred by the evaluated methods to serve as cues. In this evaluation, we only considered visits with duration of more than 10 min, as is commonly used in related work [4], [5]. Consequently, all the evaluated methods used a time threshold parameter *timeThr* set to 10 min.

We identified 92 visits from the dataset. The maximum duration of a visit was 188 min and the average was 44 min.

5.2 Methods

We compared our proposed method with three methods applied in recent works, described in Sect. 2. The settings for each method are described below. After experimenting with various input parameters, we chose the parameters that provided the best results with our dataset.

Proposed Method. We set the input parameters *extThr*, *distThr*, and *timeThr* to 40 m, 120 m, and 10 min, respectively.

DBSCAN Method. As a representative of density-based clustering methods, we selected the place-inference method that uses DBSCAN, which has been applied in several works [7], [14]. We set the distance threshold and the number of points threshold to 40 m and 40 points, respectively. GPS traces may contain time gaps because the GPS logger may stop logging when there is weak GPS signal indoors. The lack of GPS points may lead DBSCAN to an incorrect estimation of the density of a cluster. To solve that problem, we added a preprocessing step, which is also applied in previous works [7]. The preprocessing step fills the time gaps with new GPS points generated by interpolating location and time. We also added a post-processing step to prune from the result the clusters that represent visits that were shorter than the time threshold *timeThr* (10 min).

Kang Method. To represent time-based clustering methods, we selected the place-inference method proposed by Kang et al. [5]. We set the radius threshold, the time threshold *timeThr*, and the size of the buffer to 200 m, 10 min, and 2 min, respectively.

Stay Points Method. To represent radius-based clustering methods, we selected the simple and quick place-inference method applied by Zheng et al. [2], [13]. We set the time threshold *timeThr* and the radius threshold to 10 min and 200 m, respectively.

5.3 Metrics

We evaluated the inferences of each method as follows. First, for each method and trip, we matched the returned clusters with the actual visits considering the location, i.e., a visit was matched to all the clusters with an inferred location that is the visit location. We then evaluated the accuracy of the visit location inference for each trip using *precision* and *recall* metrics, which are defined as follows:

$$Recall = \frac{Mv}{Sv} \tag{4}$$

where Mv is the number of visits matched to at least one cluster and Sv is the number of visits during a trip.

$$Precision = \frac{Mc}{Sc} \tag{5}$$

where Mc is the number of clusters matched to a visit and Sc is the number of clusters computed for a trip.

Finally, we evaluated the accuracy of the visit duration inference for each visit using the *percentage of duration error* metric, defined as follows:

$$Percentage\,of\,duration\,error = 100 * \frac{|AD - ID|}{AD} \tag{6}$$

where AD is the actual duration and ID is the inferred duration of a visit.

5.4 Experimental Results

The results of the visit location inference are shown by the chart in Fig. 3. The chart shows the average recall and precision computed for each trip. Because our proposed method and DBSCAN have similar results, Figs. 4 and 5 show the micro-precision and micro-recall, i.e., the results for each GPS trace. To show the charts clearer and focus on the cases of bad inference, Figs. 4 and 5 do not show the GPS traces where precision and recall are both perfect (value 1).

The results of the visit duration inference are shown in Fig. 6. The distribution of the *percentage of duration error* computed for each visit is represented as follows. The box plot displays the interquartile range (the boxes), the median number (the horizontal lines in the boxes), and the average (the diamonds next to the values). The max and min values (whiskers) are not represented in the plot in order to show the other values clearly. The minimum value of the *percentage of duration error* was zero for all methods. The maximum value of the *percentage of duration error* was 164%,

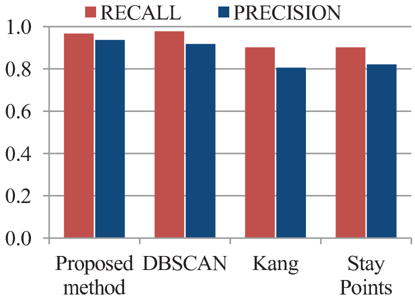

Fig. 3 Results of the visit location inference.

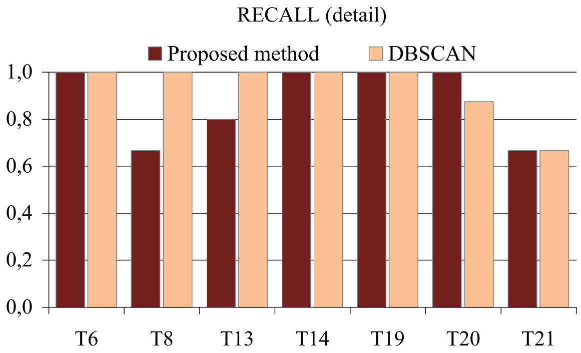

Fig. 4 Detailed recall results.

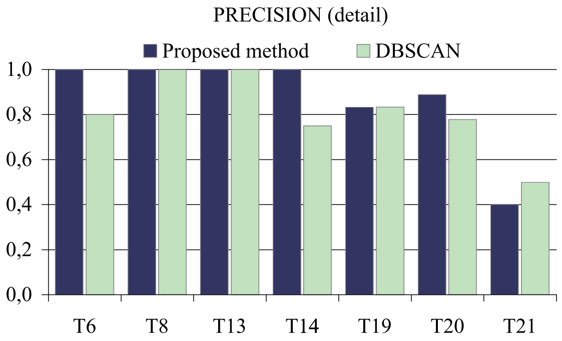

Fig. 5 Detailed precision results.

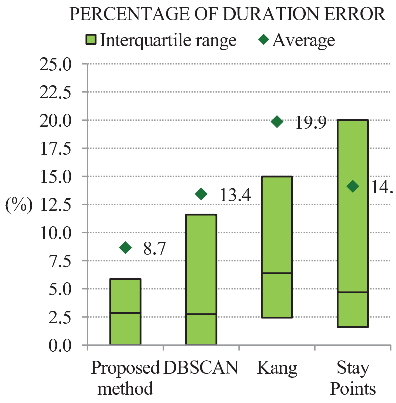

Fig. 6 Results of the visit duration inference.

Table 2 Frequency of errors caused when inferring visit locations and durations.

Error type	Bad inferences	Frequency[a]			
		Proposed method	DBSCAN	Kang	Stay Points
ER1	Bad *recall* (visit not found)	1 visit (1.1%)	2 visits (2.2%)	4 visits (4.3%)	3 visits (3.3%)
	High (> 30%) *percentage of duration error*	0 visits (0%)	0 visits (0%)	3 visits (3.3%)	0 visits (0%)
ER2	Bad *recall* (visit not found)	0 visits (0%)	0 visits (0%)	5 visits (5.4%)	3 visits (3.3%)
	High (> 30%) *percentage of duration error*	2 visits (2.2%)	4 visits (4.3%)	7 visits (7.6%)	4 visits (4.3%)
ER3	High (> 30%) *percentage of duration error*	2 visits (2.2%)	6 visits (6.5%)	3 visits (3.3%)	8 visits (8.7%)
ER4	Bad *precision* (spurious cluster)	5 clusters (5.3%)	6 clusters (6.1%)	16 clusters (15.5%)	11 clusters (10.9%)

[a] Number and percentage of visits and clusters with errors inferred by the methods.

214%, 214%, and 171% for the proposed, DBSCAN, Kang, and Stay Points methods, respectively.

5.5 Error Analysis

To understand in more detail the cases of bad inferences in our evaluation, we analyzed the errors made by the evaluated methods. We identified the following four types of errors:

Visit differentiation error (ER1): The *visit differentiation error* provides a cluster containing the GPS points logged during multiple visits.

Visit accuracy error (ER2): The *visit accuracy error* provides a cluster containing GPS points logged during a visit as well as GPS points logged right before and after the visit.

Visit completion error (ER3): The *visit completion error* provides a cluster containing only a portion of the GPS points logged during the visit.

Visit identification error (ER4): The *visit identification error* provides a cluster containing GPS points that do not seem to be a visit (e.g., crossroads, traffic lights, commutation, stops, or slopes).

The frequency of each type of error is shown in Table 2.

6. Discussion

In this section, we discuss the results of our evaluation and the main shortcomings of the methods evaluated. We also consider the two requirements discussed in Sect. 3 and the four errors identified in Sect. 5.

6.1 Method Comparison

6.1.1 Inferring Visit Locations

In Fig. 3, it can be seen that our proposed method and the DBSCAN method provide similar recall and precision, which is better (closer to 1) than the other two methods. In particular, from the results shown in Figs. 4 and 5, we can assert that our proposed method and the DBSCAN method provide better results when the GPS traces are by car, in urban areas, and with mostly outdoor visits. Moreover, in GPS traces with mostly outdoor visits, the DBSCAN method has better recall and our proposed method has better precision. We therefore state that our proposed method and the DBSCAN method can infer the locations of visits better than the other two methods.

The main reason for this is that only our proposed method and the DBSCAN method are able to generate clusters of arbitrary shapes, i.e., fulfill requirement R1. The Kang method can merge clusters to form different shapes but only those that are larger than its radius threshold. The Stay Points method can only generate clusters with a circular shape. Consequently, Table 2 shows that the Kang and Stay Points methods more frequently result in the two types of errors related to requirement R1, ER1, and ER2, causing bad recall. Our proposed method and the DBSCAN method produce ER4 errors less frequently. This is because the spurious clusters they generate are smaller and often discarded in their final step.

6.1.2 Inferring Visit Duration

In Fig. 6, it can be seen that our proposed method has a better dispersion (the interquartile range is narrower and closer to zero) and a better average *percentage of duration error* (closer to zero). Therefore, we can state that our proposed method achieves the best inference of visit duration.

The main reason for this is that only our proposed method is able to generate clusters that include far or dispersed points, i.e., best fulfill requirement R2, by applying the two rules of the defragmentation step. Consequently, Table 1 shows that the proposed method less frequently results in the error related to requirement R2, ER3, causing a high *percentage of duration error*. The Kang method produces ER3 errors in a few cases because it tends to generate large clusters that include the dispersed points. However, generating large clusters causes the Kang method to frequently produce ER2 errors, causing a high *percentage of duration error*.

6.2 Method Shortcomings

6.2.1 Bad Inferences of Visit Location

The four methods yielded poor recall when a traveler walked between two close visits. In that situation, the methods tended to generate an ER1 error because the distance and duration of the walk were short.

The four methods yielded poor precision when the traveler went slowly or even stopped for a few minutes between visits because of obstacles such as crossroads or traffic lights. In this situation, the methods tended to generate an ER4 error because the GPS points have high spatial density and low speed.

6.2.2 Bad Inferences of Visit Duration

The four methods provided a high *percentage of duration error* when inferring the durations of visits in two situations.

On the one hand, when the traveler walked to a location near a visit, the methods tended to generate an ER2 error. This is because GPS points unrelated to visits were close to a visit. This situation causes a high *percentage of duration error*, especially when the visit duration is short (e.g., less than 30 min).

Conversely, when the traveler walked quickly during a visit across a large location (e.g., a park), the methods tended to generate an ER3 error because the points become dispersed.

6.3 Visit Location Size versus Duration

One other notable observation from our evaluation is that bad inference of the size of a visit location does not always imply bad inference of the visit duration. For example, a method may infer that the region of a visit location is twice the size of the actual region, but it may infer the visit duration with only a few seconds of error. The visit duration is inferred with a small error when travelers move quickly (e.g., by car) outside the visit locations. In this case, a few hundred meters can be traveled in less than a minute.

7. Conclusions

In this paper, we presented a pace-based clustering method to infer the visits during a trip from its GPS trace. We argued that inferring the duration of each visit is essential to answering queries for travel recommendation and travel diary generation applications. Our proposed method contributes two novel techniques for inferring visit location and duration. The techniques allow the method to generate clusters of arbitrary shapes, omitting points that are close but unrelated to visits, and clustering far or dispersed points logged during visits. Evaluation of our proposed method revealed two main results: (1) Our proposed method infers a visit duration with an average error of 8.7%, while the best of the three evaluated existing methods inferred visit duration with an average error of 13.4%. (2) Our proposed method inferred a visit location with a recall and precision comparable to the best of the other three evaluated methods.

In future work, we plan to apply various kinds of map data to avoid the shortcomings detected in the evaluated methods. Using polygon map data that represents the delimited perimeters of buildings and parks may help to distinguish the GPS data logged during the same visit and different near visits. Conversely, road and transportation map data may help to distinguish actual visits from situations that appear to be visits (e.g., time spent at crossroads and traffic lights).

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] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft, "Recommending social events from mobile phone location data," Proc. 10th IEEE Int. Conf. on Data Mining (ICDM), pp.971–976, 2010.

[2] Y. Zheng and X. Xie, "Learning travel recommendations from user-generated GPS traces," ACM Transactions on Intelligent Systems and Technology, vol.2, no.1, pp.1–29, 2011.

[3] J. Wolf, R. Guensler, and W. Bachmann, "Elimination of the travel diary: An experiment to derive trip purpose from GPS travel data," Transportation Research Record, vol.1768, pp.125–134, 2001.

[4] D. Ashbrook and T. Starner, "Using GPS to learn significant locations and predict movement across multiple users," Personal and Ubiquitous Computing, vol.7, no.5, pp.275–286, 2003.

[5] J. Kang, W. Welbourne, B. Stewart, and G. Borriello, "Extracting places from traces of locations," Proc. 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN hotspots (WMASH), pp.110–118, ACM Press, 2004.

[6] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personally significant places: An interactive clustering approach," ACM Transactions on Information Systems, vol.25, no.3, Article 12, 2007.

[7] B. Adams, D. Phung, and S. Venkatesh, "Extraction of social context and application to personal multimedia exploration," Proc. ACM Conference on Multimedia (MM), pp.987–996, ACM, 2006.

[8] R. Aipperspach, T. Rattenbury, A. Woodru, and J. Canny, "A quantitative method for revealing and comparing places in the home," Proc. 8th International Conference on Ubiquitous Computing (UbiComp), LNCS, vol.4206, pp.1–18, Springer, 2006.

[9] P. Nurmi and S. Bhattacharya, "Identifying significant places: The non-parametric way," Proc. 6th International Conference on Pervasive Computing (Pervasive '08), J. Indulska, D.J. Patterson, T. Rodden, and M. Ott (eds.), pp.111–127, Springer-Verlag, Berlin, Heidelberg, 2009.

[10] P.M. Lerin, D. Yamamoto, and N. Takahashi, "Inferring and focusing areas of interest from GPS traces," Proc. 10th International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2011), LNCS 6547, pp.176–187, Springer, Kyoto, Japan, March 2011.

[11] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, and J. Hughes, "Learning and recognizing the places we go," Proc. 7th International Conference on Ubiquitous Computing (UBICOMP), LNCS, vol.3660, pp.159–176, Springer-Verlag, 2005.

[12] N. Marmasse and C. Schmandt, "A user-centered location model," Personal and Ubiquitous Computing, vol.6, no.5–6, pp.318–321, 2002.

[13] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location history," Proc. 16th ACM SIGSPA-

TIAL International Conference on Advances in Geographic Information Systems (GIS '08), Article 34, pp.1–10, ACM, New York, NY, USA, 2008.

[14] M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu, "Incremental clustering for mining in a data warehousing environment," Proc. 24th International Conference on Very Large Data Bases (VLDB), pp.323–333, Morgan Kaufmann Publishers, 1998.

[15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Proc. International Conference on Knowledge Discovery and Data Mining (KDD), pp.226–231, AAAI, 1996.

[16] J.T. Lehikoinen and A. Kaikkonen, "PePe field study: Constructing meanings for locations in the context of mobile presence," Proc. 8th Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'06), pp.53–60, ACM, 2006.

Pablo Martinez Lerin earned a five-year B.E. degree from the Polytechnic University of Valencia (Spain) in 2009, and a Ph.D. in Computer Science and Engineering from Nagoya Institute of Technology (Japan) in 2013. He is a researcher at Yokohama Research Laboratory in Hitachi Ltd. (Japan). His research interests include user experience design, IT management systems, and collaborative GIS systems.

Daisuke Yamamoto is an associate professor in the Information Technology Center at Nagoya Institute of Technology, Japan. He received a Ph.D. in Information Science from Nagoya University. His research interests include Web services, Web interaction, content technologies, E-learning, GIS, and multimedia systems.

Naohisa Takahashi is a Professor in the Department of Computer Science at Nagoya Institute of Technology, a position he has held since 2001. Prior to coming to NIT, he was engaged in research on parallel processing, software engineering, and network computing at NTT Laboratories for 25 years. He received B.E. and M.E. degrees in Electrical Engineering from the University of Electro-Communications, Tokyo, Japan, in 1974 and 1976, respectively. He also received a doctorate in Computer Science in 1987 from Tokyo Institute of Technology. His recent research interests are network computing, ubiquitous computing, geographical information systems, and e-learning systems. Dr. Takahashi is a member of the IEEE, the Association for Computing Machinery, the Information Processing Society of Japan, the Japan Society for Software Science and Technology, and the Database Society in Japan.