Vol.2016-MBL-80 No.16
Vol.2016-CDS-17 No.16
2016/8/25

Train Route Detection with using Sensor Data and GIS

Masaki Ono[1,a)]

Abstract: Smart phones have many sensors (GPS/pedometer), so data can easily be obtained to estimate a user's context, which is an important factor for smart phone applications. In this paper, we present a method to detect past train routes (a sequence of stations) from GPS records using station location information. Although GPS sensors do not work well underground or in overcrowded areas, we solve this problem by thinking through all possible paths and weighting them using sensor records. To evaluate our method, we conducted an experiment with sensor data created by several people over more than two weeks and found that its accuracy exceeded the baseline accuracy.

1. Introduction

Smart phones have many sensors such as GPS an pedometer, so data can easily be obtained to estimate a user's context, which is an important factor for smart phone applications. A common application is transport mode detection, which is a method to determine how a user travels, such as walking, driving, and taking trains.

In this paper, we present a method to estimate train routes a person has used from a sequence of sensor data and station location information. If an organization can know exactly which trains employees use, it can easily check their expense reports or optimize train routes.

The aim of the proposed method is to detect train routes (a sequence of stations) from GPS records using station location information. We show the task overview in figure 1. The sensor data is categorized into 2 kinds, location data and step data. Here, station location information consists of a set of latitude and longitude pairs that indicate centers of stations. We do not use train track information because it is released by very few organizations, such as the governments of Japan and Chicago, USA.

A sequence of stations a person used is not easy to precisely detect from GPS records and station location information because of the following problems.

(1) Many stations are closely located in urban areas, so there are a number of candidate stations.
(2) GPS sensors do not work well underground, so there is a lack of GPS records for stations located underground.
(3) GPS sensors do not work well in overcrowded areas due to network capacity overflow.

Subways are common in major cities, and even nominally overground train lines sometimes have stations located underground. Additionally, a big station has multiple train lines and many people passing through it. Consequently, it sometimes becomes an overcrowded area, so GPS sensors have a high probability of failing to capture data.

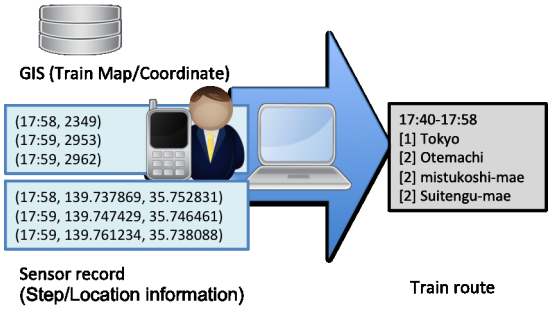

Fig. 1 Overview of the task

The proposed method handles a sequence of sensor data and station location information and estimates train routes a person has used. The proposed method consists of three steps. The first step extracts the duration for which a person took a train from a sequence of sensor data. In the next step we detect a train route from each of the durations by considering all possible paths by using a train map. Here, we give a candidate path a confidence score by comparing sensor data and evaluating reasonability. Finally, we think the reasonability of all train routes again and update them if the reasonability is not high. These steps enable missing stations to be found.

To handle stations that are closely located, the proposed method uses a variable threshold for evaluating the distance between a station and a person. The threshold is determined by station size, which is defined as the number of train lines at the station.

To evaluate our method, we conducted an experiment with sensor data created by four people over more than two weeks. The results showed the method identified missing stations better than conventional methods. We also tested several functions for evaluating the reasonability of a train route candidate.

2. Motivation

From technical perspective, we have two issues for handling

[1] IBM Research - Tokyo, 19-21, Nihonbashi Hakozaki-cho Chuo-ku, Tokyo 103-8510 Japan
[a)] moono@jp.ibm.com

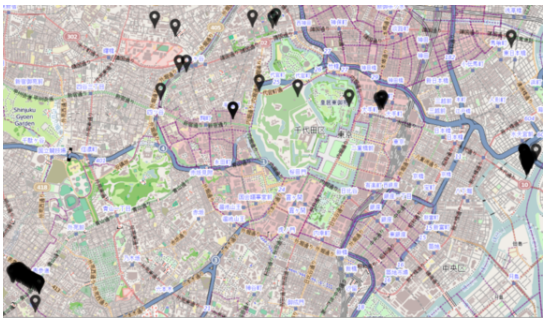

Fig. 2 Example of location information

GPS and station location information. At first, Since stations are nearly located in the urban, we have many candidate stations which can be consists of a train routes. For example, we used the location information distributed in the web site "駅データ.jp"[*1], and found that more than 3,188 station pairs are located within 500 meters in Japan. Although there are 9,807 train stations in Japan, the he number of these stations are not many, it is not small issue when we think abut the number of people who usually use train. Train is common transport for people in the urban area.

The second issue is that GPS does not work well underground and in overcrowded area. Other research work reports the same issue [1][2], and applied methods with other sensors such as barometer or acceleration.

The figure 2 shows the points reported by GPS sensor when a person took subway from *Suitengumae* to *Omotesando*. Here we do not care about accuracy and use all location information we captured. We do not capture location information around *Suitengumae* and during the trip. If the network status is fine, we can capture location information at intervals of 20 meters. This issue also occurs in overcrowded area due to the exceed of the network capacity. In other words GPS sensor often does not create reports during the rush hour.

It can be a crucial issue for creating valid candidate paths from train map that we do not capture start/end stations or almost all part of train routes. However, it is not a big issue that we do not capture a part of train routes. Because we can complement it by using context before/after the lack.

From business perspective, we have two reasons for the work. Generally understanding user context is key factor to make services more user-friendly.

The second reason is that we can check an employee's expense easily if we combine the proposed method with an application in the employee's mobile device. It is useful to reduce time for dealing with his/her expense and errors in procedures about expense. Since many people use train in urban areas, it is not a small issue.

3. Definition of Sensor Data/ Geographical Information System (GIS) Data

We use a sequence of sensor data to estimates the train routes the device's user has taken. Sensor data can be categorized into two kinds: location data (L), which is mainly created from GPS records, and step data (S), which is created from pedometer records.

[*1] http://www.ekidata.jp/

Table 1 The example of sensor data

time	*longtitude*	*latitude*	*accuracy*	*speed*	*step*
17:58:56	35.67892322	139.7872693	20	-1.0	N/A
17:58:58	N/A	N/A	N/A	N/A	2949
17:59:00	N/A	N/A	N/A	N/A	2953
17:59:03	N/A	N/A	N/A	N/A	2958
17:59:05	N/A	N/A	N/A	N/A	2962
17:59:15	35.67913860	139.7864844	20	5.0	N/A
17:59:18	35.67896539	139.7868399	20	7.5	N/A

Let $L = \{l_1, l_2, ..., l_n\}$ be a sequence of location information. Location information l_i consists of five elements. Note that this data model is common and has been used in another work [3].

- $time_i$ represents a timestamp of sensor data l_i
- p_lat_i represents latitude of device
- p_lon_i represents longitude of device
- $speed_i$ represents current ground speed of device
- $accuracy_i$ represents accuracy of positioning

Let $W = \{w_1, w_2, ..., w_n\}$ be a sequence of step information. Step information w_i consists of two elements.

- $time_i$ represents a timestamp of sensor data w_i
- $step_i$ represents number of steps

We use two kinds of GIS data: station location data and train map data. Let $S_{all} = \{s_1, s_2, ..., s_n\}$ be the set of station data. s_i consists of four elements:

- s_lat_i represents latitude of a station
- s_lon_i represents longitude of a station
- s_name_i represents name of a station
- s_line_i represents line on which a station is located

A train map is a graph that represents connectivity among stations. Let E_{all} be a set of vertexes where each element represents connectivity of a pair of stations. Then we can define a train map as: $G = (S_{all}, E_{all})$

Table 1 shows examples of L and S, which is generated by iPhone6. Since L and S are created independently, their timestamps are different. Correctly, location sensor of iPhone6 uses not only GPS but also other information sources[*2]. Accuracy indicates the radius of uncertainty for the location.[*3] Negative value in the speed column indicates measurement error.

4. Related work

Transport mode detection is one of the most common tasks in wearable sensor data analysis and there are many work. GPS sensor is frequently used in transport mode detection [4][5] . For example, Zheng et al. [4] studied transport mode detection using GPS sensor records. This work handles 4 modes (car, bus, bike, and walk) and train is not handled.

Stenneth et al. [3] used not only GPS records but also GIS, train station information, train track information, bus stop location information, and bus operation status information. They conducted their experiment on information made public by the City of Chicago, USA through its website. To distinguish cars from trains, GIS is very important.

There are some work which focuses on acceleration sensor to

[*2] https://support.apple.com/en-us/HT203033
[*3] https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CLLocation_Class/index.html

know a user's context. Hemminki et al. [6] and focused on acceleration sensor records and achieved high precision/recall in detecting six modes of transport. An acceleration sensor is one of the most important devices for understanding a person's context. Dernbach et al. [7] reported that their acceleration-based method precisely recognizes common activities such as riding a motorbike or driving a car.

Handling location information reported by a sensor on subway is a difficult task. Jimbo and Fujinami [1] focused on an acceleration sensor for subway route detection. They predicted the direction of movement a user taken from acceleration sensor records and combined it with a train map. Stockx et al. [8] also focused on an acceleration sensor to know location of a subway. Hyuga et al. [2] calculated altitude from barometer records and focused on changes in altitude for subway route recognition. Since subway stations are not located at the same altitude even when they are on the same line, a sequence of altitude information is a useful feature set for route recognition. Thiagaraja et al. [9] uses acceleration sensor and GPS sensor because GPS is well known to drain the battery on a smartphone quickly. To handle the lack of location information HMM is used in the work.

Map matching is a similar task to our work and the aim is to identify the correct road on which a car runs. This is one of the most common tasks which focus on GPS sensor records. Quddus et al. [10] reports there are at least 35 algorithms during the period 1989-2006. Some work focuses on the lack of location information or low-sampling-rate GPS. Although we do not use train track information and the task setting is not equal, the motivation is similar.

5. Method

The proposed method consists of three steps, and an overview is shown in Figure 3. The first step is transport mode detection: the duration for which a person took a train is detected from a sequence of sensor data. The second step is route detection: a sequence of stations a person used in the detected duration is detected from the same sequence of sensor data. The final step is route refinement: the train route generated in the previous step is updated by considering reasonability of a sequence of train routes. In the route detection step, we remove a duration we can not create any reasonable train routes. Therefore, correctly transport mode detection is not conducted only in the first step.

5.1 Transport mode detection

Before considering the train route, we detect a duration for which a person took a train from a sequence of sensor data D. We receive D as input and create a set of index number pairs that represent the start and end of a duration for which a person might have taken a train as output.

As discussed in section 4, most related work has used statistical approaches. In contrast, we use a rule-based approach for three reasons. The first reason is that the duration for which a person took a train is not difficult detect. Because there are not many kinds of vehicles which can move at high speeds. The second one is that A rule-based approach do not need training data and is easy to start. The third one is that we can understand what

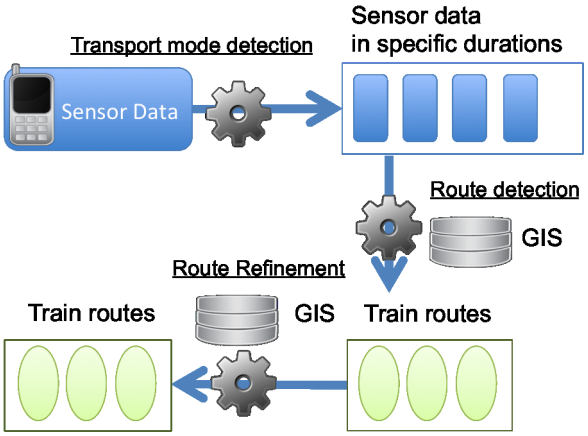

Fig. 3 Overview of the task

is wrong rather than considering the result of machine learning.

In this section we do not use GIS and focus on the sensor data. This section extracts a duration for which a person move at high speeds. Since not only train but also car can run rapidly, we distinguish cars from trains by using GIS and considering a train route.

We have two ways to know the device speed. One way is to see the speed reported by the sensor. The other way is to calculate using location information and timestamp in the sensor data. Here we combine the two ways for improving the accuracy of the value.

The first phase is smoothing. Since sensors do not update values in the same interval and these values naturally contain errors, we have to use the frame to handle them correctly.

We calculate a sequence of maximum speeds per minute MS from a sequence of location information L. Let L_{min} be a set of speed values measured during a specific time min. Then the maximum value in the L_{min} is defined as the maximum speed per minute ms_{min}.

After the smoothing phrase, we pick out a duration for which a person moved at high speed. Let MIN be a sequence of minutes that represents a duration when both sensors worked. Then we define the duration using the threshold T_{speed}:

$$[min_s, min_e] = [min | min \in MIN, ms_{min} > T_{speed}] \quad (1)$$

We calculate a sequence of a set of stations for a duration for which a user rode a train. Let $L_{se} \in L$ be a sequence of location data measured during $[min_s, min_e]$. Let $near$ be a function that maps location data l_i to a set of stations that represents stations near l_i. Then we define a sequence of station sets $S_{observed}$:

$$S_{observed} = near(l_i) \rightarrow near(l_{i+1}) \rightarrow \ldots \quad (2)$$

Function $near$ calculates the distance between l_i and s_i and then determines whether a station s_i is near or not by comparing distance with the threshold. One of our contributions is to put a variable threshold in $near$. A big threshold catches unnecessary stations when stations are closely located, and a small threshold does not catch correct stations when a platform is too far from the center of s_i. To solve this problem, we put a threshold that reflects the size of s_i.

Table 2 Example of a sequence of distance for the nearest station

Time	Station	Dist. (m)
17:49	*Ikebukuro*	363.63
17:52	*Ikebukuro*	608.09
17:53	*Kita-Ikebukuro*	561.69
17:54	*Kita-Ikebukuro*	558.1
17:56	*Sugamo-Shinden*	639.37
17:57	*Nishi-Sugamo*	526.73
17:58	*Nishi-Sugamo*	247.11
17:59	*Nishi-Sugamo*	79.74
18:01	*Nishi-Sugamo*	252.26
18:03	*Nishigahara-Yotyome*	243.7
18:05	*Asukayama*	119.12
18:06	*Asukayama*	174.24
18:07	*Oji-ekimae*	55.152

With a sequence of a set of stations $S_{observed}$, we determine whether a person took a train during $[min_s, min_e]$. If we find any reasonable paths a person might have used on the train map G by using $S_{observed}$, we consider the duration $[min_{start}, min_{end}]$ to be a train. Otherwise, we consider it to be a car. In the next section, we discuss how to find a reasonable path with sensor data and GIS.

Table 2 shows the nearest station and distance to it created from sensor records that are considered to be of a person using a car. The possibility of train use is naturally low because we cannot think of any reasonable paths on a train map that match these records. In fact, these sensor records are from a person riding a bus.

5.2 Route detection

In the previous section, we extracted a set of durations for which a person might have taken a train. Here, we predict all train routes that consist of a sequence of certain stations. First, we detect candidates for start and end station pairs. Then we calculate paths that connect these start and end stations on a train map G_{all}. Finally, we rank these paths and detect the most reasonable one.

In the final phase, we conducted two kinds of ranking strategies: one to handle each train route independently, and the other to handle all train routes at the same time. A person in an urban area often uses more than one train line to go to his/her destination.

We do not use train track information. Instead, we use a set of latitude and longitude pairs that indicate centers of stations and consider the possibility of train use by using a train map and observed stations.

There is a set of durations D, and candidate start and end station pairs are detected in one duration $d \in D$. The general idea is to assume stations observed at the start/end of a duration to be start/end stations.

Let $m1$ and $m2$ be given numbers used in the candidate station search. Then we define start station candidates S_{start} and end station candidates S_{end} in d.

$$S_{start} = \bigcup_{i=start-m1}^{start+m2} near(l_i) \tag{3}$$

$$S_{end} = \bigcup_{i=end-m1}^{end+m2} near(l_i) \tag{4}$$

We calculate products between S_{start} and S_{end} and define candidates of train routes R as simple paths between elements of the products. Let $spath$ be a function which maps station pairs and a graph into a set of simple paths on the graph.

$$\begin{aligned} R = \{r | & s_s \in S_{start}, \\ & s_e \in S_{end}, \\ & r \in spath(s_s, s_e, G)\} \end{aligned} \tag{5}$$

We select the most reasonable train route from a set of train routes R. To weight a train route $r \in R$, we calculate a confidence value for each r. If the highest confidence value is less than the threshold, we conclude that there are no paths and that the person did not take a train. This is a part of transport mode detection.

To discuss a feature set that comes from timestamps of location information, we define $stime$ as a function that maps station s_i to a set of $time_i$ when s_i is observed. The following are feature sets we used for calculating a confidence value of r.

Matching ratio This represents the similarity between a candidate route r and a set of observed stations $S_{observed}$. This is defined as $\frac{|r \cup S_{observed}|}{|r|}$.

Duration This represents the size of a range from when start station s_s is first observed to when end station s_e is last observed. This is defined as $\arg\max stime(s_s) - stime(s_e)$.

Shortest path This represents the shortest path length between a pair of stations s_s and s_e.

5.3 Route Refinement

After the procedures described in the previous section, we have a sequence of train routes R_{all}. Here we think the reasonability of each routes again. We focus on moving speed from one route to the next route. In other words, if current train routes R_{all} indicates that a person walked with high speed, we assume that R_{all} is not correct.

We calculate a sequence of the number of steps per minute NS from a sequence of step information S. Let S_{min} be a set of step data measured during a specific time min. Then the averaged value of S_{min} is defined as step per minute ns_{min}.

By using NS and L, We can know the location where a person took a walk just before a user took train. If we do not find any locations, the station which is the end of R_{i-1} is picked up.

We pick up stations which can connect to the start station of R_i. Then, for each of the stations we calculate distance from the location, and we calculate moving speed between these two locations. Since both the start station of R_i and the location have the timestamps, we can calculate the speed.

If we find a station which is more reasonable than the first station of R_i, we assume the station is first of R_i. And we complement the train route between the previous first station and the current first station of R_i by using train map G.

The procedure above handles the start station of a train route. We also apply the same kind of procedure for the end station of a train route and update it if we find more reasonable stations.

The figure shows the example of the procedure in the route refinement. Although we have sensor data that indicates routes R_1 and R_2, it is not possible to walk from the station 2 to the station

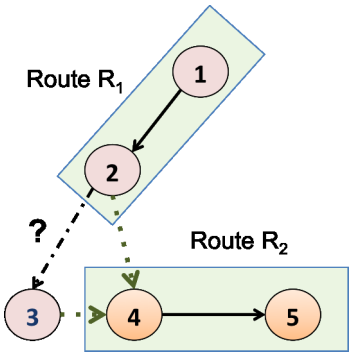

Fig. 4 Overview of the task

4, because there is not enough time. Therefore we can think that the end of R_1 is not station 2 but station 3.

6. Evaluation

In the section we evaluate the proposed method and characteristics. Precisely the aim of the evaluation is to answer the following questions:

- What feature set is the best for selecting correct train routes?
- Did simple transport mode detection work well?
- Did the variable threshold catch missing stations?
- Is train route refinement effective for catching missing?

To conduct the experiment, we created the mobile application which can collect sensor data and obtained 139 train routes produced by several persons over more than two weeks. The correct train routes are manually defined by a persons who rode on the train. And we used the station information and the station map distributed in the web site “ 駅データ.jp ”.

We set the following parameter values in the experiment: T_{speed} = 30, m_1 = 3, m_2 = 3. Let *next* be a function which maps a station and a graph to a set of stations connected to the given station on the graph. Then we define the threshold in *near* function as $160 + 40 \times next(s_i)$. It means that the threshold depends on the number of platforms and linearly increases.

What feature set is the best for selecting correct train routes? Here we think what feature set is effective for ranking train routes in the task. To evaluate each of feature sets, we simply sort train routes using one or two feature sets. If there are several train routes which have the save value on the first feature set, we compare them by using the second feature set. The table **?** shows the result of the ranking functions. Duration is the most effective in the these 4 feature sets. And the combination of duration and matching achieves the highest accuracy.

Shortest path is not so effective in the experiment. Here we took variety kinds of train routes to evaluate the method from many aspects. Therefore a train route in the experimental data is not always shortest. For example, we selected more than two kinds of routes to go to the *Futamatagawa* from *Shibuya*. We do not understand whether this problem occurs only in the experiment or not.

It is not surprising that the matching ratio does not work well for subway route detection. Because in the task we do not observe the stations a person passed. Therefore we can conclude that matching ratio works for trains running on ground.

Table 3 Accuracy and feature sets for ranking train routes

1st key	2nd key	accuracy
shortest path		0.568
duration		0.698
matching ratio		0.482
duration	shortest path	0.691
matching ratio	duration	0.770
duration	matching ratio	0.777
matching ratio	shortest path	0.489

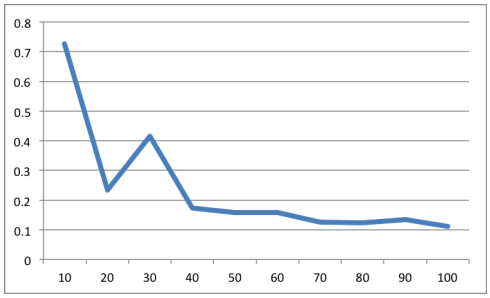

Fig. 5 Speed and absolute ratio of the two kinds of speeds

In this work we use graph where each of vertexes does not have a weight and do not use train timetable. If we use train timetable, we can create more feature sets. Since trains at the rash hour often do not work under timetable, there is the potential that train timetable brings another problem.

Did simple transport mode detection work well? The proposed method detects the transportation mode of long trips with high accuracy . If a train route consists of two or three stations, the proposed method do not understand the transportation mode. For example, it takes about 1 minute from *Tokyo* to *Otemachi* and the distance between two stations are 500 meters. It is difficult to know whether a person rode on train or walk by focusing only on speeds. To handle short trips, we have to focus additional information source such as barometer or acceleration.

The data contains more than four drive tracks created by several people, and the proposed method can distinguish them completely. If a car runs on a road beside a train track, it is difficult for the proposed method to distinguish whether a person is driving a car or riding a train, however related work with only GPS/GIS data obviously has the same problem. The problem does not appear in urban areas because there are almost no roads that are long and beside a train track.

We can predict the moving speed in the two ways, and they are not so different if the speed is more than 40 km/h. The first way is to use values reported from a GPS sensor. And the other way is calculates from reported locations. Figure 5 shows the relationship between sensor reported speed and the absolute ratio of the two speed. The values which is lower than 10 km/h has large difference, we do not put on the figure. Generally GPS's error is limited less than 10 meters when we walk on ground. However, it is a big issue for calculating low speeds.

Did the variable threshold catch missing stations? The proposed method search stations around a given location information with *near* function and uses the variable threshold in *near* function. To evaluate how the variable threshold works, we applied the fixed threshold in *near* function.

Table 5 shows thresholds in *near* function and accuracy. We

Table 4 threshold in *near* function and station detected as start of train route

Threshold	Company	Station
proposed	JR East	*Yurakutyo*
	JR East	*Shinbashi*
	Sagami Railway	*Yokohama*
	JR East	*Shibuya*
400m (fix)	Tokyo Metro	**Ginza**
	JR East	*Shinbashi*
	Sagami Railway	*Yokohama*
	JR East	*Shibuya*
200m (fix)	JR East	*Yurakutyo*
	JR East	*Shinbashi*
	Sagami Railway	*Yokohama*
	JR East	**Ebisu**
100m (fix)	JR East	*Yurakutyo*
	JR East	*Shinbashi*
	Sagami Railway	**Hiranuma-bashi**
	JR East	**Ebisu**

Table 5 threshold in *near* function and accuracy

Threshold	Accuracy
proposed	0.770
100m (fix)	0.655
200m (fix)	0.741
300m (fix)	0.741
400m (fix)	0.719

use the combination of matching ratio and duration as feature sets in the evaluation. The variable threshold achieves the highest accuracy.

Table 4 shows thresholds in *near* function and stations detected as starts of train routes. The stations with bold style are wrong in the table. If we use a fixed value, we do not correctly observe start/end stations, and the accuracy of the method decreases. The big fixed threshold catches unnecessary stations when stations are closely located, and the small fixed threshold does not catch correct stations when a platform is too far away from the center of a station.

The number of station pairs which are located closely is not big. And the pairs are mainly located in urban areas. Therefore the effect of the variable threshold is limited when we think only the number of stations. However many people live or work in urban area, we believe that this is useful for the task.

Is train route refinement effective for catching missing stations? To evaluate train route refinement, we compare the method with train route refinement and the method without train route refinement. Here we use the combination of matching ratio and duration as feature sets.

The method with train route refinement extracts 107 (77.0%) correct train routes and the method without refinement extracts 93 (67.0%) correct train routes. The accuracy is increased by 10 points. We can conclude that train route refinement is effective for the lack of location information.

It is difficult to capture location information in subway, however the lack occurs on big stations. In the dataset GPS sensor does not work well in *Yokohama* station at the rash hour frequently. Therefore train route refinement is useful for not only subway but also all trains.

7. Conclusion

In this paper, we proposed the method to detect train routes (a sequence of stations) from GPS records using station location information If an organization can know exactly which trains employees use, it can easily check their expense reports or optimize train routes.

We have two issues for handling GPS and station location information. At first, since stations are nearly located in the urban, we have many candidate stations which can be consists of a train routes. To solve the issue, we used the variable threshold when we consider stations nearly located at the given location information. The second issue is that GPS sensors do not work well underground and in overcrowded areas, so there is a lack of GPS records. To handle the problem we think the reasonability of all train routes after considering a train routes singly.

To conduct the experiment, we created the mobile application which can collect sensor data and obtained 139 train routes produced by several persons over more than two weeks. In the experiment we found the variable threshold extracts stations correctly. And we found that it is effective for a lack of GPS records and the best accuracy is 0.77, which is 10 points higher that the method without train route refinement.

Future work is to use additional information which comes from sensors in a mobile device. We believe that various kinds of information improve the accuracy of the task.

References

[1] Jimbo, T. and Fujinami, K.: Detecting mischoice of public transportation route based on smartphone and GIS, *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, ACM, pp. 165–168 (2015).

[2] Hyuga, S., Ito, M., Iwai, M. and Sezaki, K.: Estimate a User's Location Using Smartphone's Barometer on a Subway, *Proceedings of the 5th International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '15, New York, NY, USA, ACM, pp. 2:1–2:4 (online), DOI: 10.1145/2830571.2830576 (2015).

[3] Stenneth, L., Wolfson, O., Yu, P. S. and Xu, B.: Transportation mode detection using mobile phones and GIS information, *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, pp. 54–63 (2011).

[4] Zheng, Y., Liu, L., Wang, L. and Xie, X.: Learning transportation mode from raw gps data for geographic applications on the web, *Proceedings of the 17th international conference on World Wide Web*, ACM, pp. 247–256 (2008).

[5] Patterson, D. J., Liao, L., Fox, D. and Kautz, H.: Inferring high-level behavior from low-level sensors, *International Conference on Ubiquitous Computing*, Springer, pp. 73–89 (2003).

[6] Hemminki, S., Nurmi, P. and Tarkoma, S.: Accelerometer-based transportation mode detection on smartphones, *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, ACM, p. 13 (2013).

[7] Dernbach, S., Das, B., Krishnan, N. C., Thomas, B. L. and Cook, D. J.: Simple and complex activity recognition through smart phones, *Intelligent Environments (IE), 2012 8th International Conference on*, IEEE, pp. 214–221 (2012).

[8] Stockx, T., Hecht, B. and Schöning, J.: SubwayPS: Towards smartphone positioning in underground public transportation systems, *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, pp. 93–102 (2014).

[9] Thiagarajan, A., Biagioni, J., Gerlich, T. and Eriksson, J.: Cooperative transit tracking using smart-phones, *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ACM, pp. 85–98 (2010).

[10] Quddus, M. A., Ochieng, W. Y. and Noland, R. B.: Current map-matching algorithms for transport applications: State-of-the art and future research directions, *Transportation research part c: Emerging technologies*, Vol. 15, No. 5, pp. 312–328 (2007).