Car-level Congestion and Position Estimation for Railway Trips Using Mobile Phones

ABSTRACT
We propose a method to estimate car-level train congestion using Bluetooth RSSI observed by passengers’ mobile phones. Our approach employs a two-stage algorithm where car-level location of passengers is estimated to infer car-level train congestion. We have learned Bluetooth signals attenuate due to passengers’ bodies, distance and doors between cars through the analysis of over 50,000 Bluetooth real samples. Based on this prior knowledge, our algorithm is designed as a Bayesian-based likelihood estimator, and is robust to the change of both passengers and congestion at stations. The car-level positions are useful for passengers’ personal navigation inside stations and car-level train congestion information helps determine better strategies of taking trains. Through a field experiment, we have confirmed the algorithm can estimate the location of 16 passengers with 83% accuracy and also estimate train congestion with 0.82 F-measure value in average.

Author Keywords
Train congestion; positioning; mobile sensing; Bluetooth

ACM Classification Keywords
H.4.2. Social Issues: Miscellaneous

INTRODUCTION
In very populated cities like Tokyo, Beijing, Paris, London and New York, train and metro systems have been well-developed and they are often congested depending on time and lines. Accordingly, using trains becomes harder for pregnancy, people with disabilities, parents with infant and others.

Live congestion information with sufficient precision (i.e. car-level congestion information) must be beneficial for those people. For example, those cars stopping close to main stairs at a huge station are tend to be more crowded than the others of the same train. It is able to directly grasp such car-level congestion if dedicated sensors such as weight sensors are installed in each car. In fact, some trains are equipped with such sensors [21], but availability is still very limited due to its infrastructure cost. Some service providers like Jorudan[19] and NAVITIME[29] that provide route information of public transportation in Japan, have already incorporated crowd-sourcing (or participatory sensing) systems to collect congestion status. However, they need to expect users’ spontaneous activities to “describe” their context (which trains they take (i.e. train identification), which cars they are (i.e. car-level position) and congestion levels) and to inject the information into the system. These should be automated to attract more people to participate in the services.

Train identification has been done by recent work on public transportation travel estimation [4], and we may rely on these technologies. On the contrary, neither car-level position estimation nor car-level congestion estimation has been considered in the past. Even with WiFi APs, radio-based position estimation becomes a challenging task on trains. One reason is that radio signals attenuate due to both distance and human bodies, where the following two situations are hard to be distinguished – a node is located far from an AP in a non-crowded car and it is located close to the AP in a crowded car.

As for congestion estimation, some existing approaches[22, 23, 35] have targeted node density estimation using information from mobile phones such as Bluetooth RSSI and acceleration. However, they are not directly applicable to train cases where node mobility is special and only a few Bluetooth devices are visible in each car. Therefore, we need a new approach to train situation recognition, by learning and fully leveraging those train-specific features.

In this paper, we propose a novel method to estimate car-level congestion and location. The proposed method is basically a participatory sensing system in which participating passengers collect Bluetooth signals emitted from nearby passengers. Through the analysis of over 50,000 real Bluetooth samples, we have built RSSI functions which determine the likelihood of two devices being in the same car (same-car confidence) and the likelihood of congestion between them (congestion confidence). Utilizing these functions as prior knowledge, we design a two-stage algorithm where car-level location of passengers is estimated to infer car-level train congestion, based on a Bayesian-based likelihood estimation and update. Furthermore, considering such train specific mobility features that most passengers on-board do not usually change their positions and not all the passengers get off the train at a station, we design a new algorithm that employs continuous update of those likelihoods over multiple station stops.

We have evaluated the proposed method using the real data collected by 16 student volunteers in 4 different lines with over 60 different stations operated by different companies in Osaka. Through the experiments, we have confirmed that the proposed system could estimate the car-level positions of those 16 people with 83% accuracy and car-level congestion with 0.82 F-measure value.

RELATED WORK
Researches for Intelligent and Smart Urban Mobility
The recent trend of urban ubiquitous services has increasingly attracted attention to smart traffic and urban mobility. In particular, location services for trains, trams, metros, and buses have been widely deployed in many countries [31, 30], which enable passengers to recognize timetables and diagrams, and choose appropriate lines and transit. In addition, Refs.[4, 38, 37, 3, 12] present tracking and arrival time estimation techniques for buses and trains by using mobile phones. Interestingly, Ref.[38] provides a unique idea of estimating passengers who have just got on-board, by detecting beep sound emitted by IC card readers. As explained in Section 1, crowd-sourcing systems have been incorporated into Jordan [19] and NAVITIME [29, 1]. For smarter mobility at transport infrastructure terminals such as airports and huge train stations, phone-based pedestrian navigation systems have been in service (e.g. Copenhagen [7] and Paris [32]).

**Congestion Monitoring and Crowd Sensing**

Several approaches to people crowd sensing, tracking and estimation have been investigated in recent years. Vision-based pedestrian detection and tracking has been already in market [27] and well-investigated for long years [11, 33]. Those technologies can be applied to intelligent transportation systems as well, e.g. computer vision algorithms for monitoring vehicles, individuals, and crowds have been proposed in Ref.[28]. However, it is not straightforward to directly apply those techniques to train infrastructure. For example, many CCTV's have already been installed on railway platforms, but it is not allowed to use them for other than safety and security purposes. Furthermore, they are not installed in passenger cars due to privacy concerns and less cost-benefit performance. In addition, we note that some modern passenger cars are equipped with weight sensors (air springs) for adaptive control of trains toward variable load [21]. Such information can of course be useful if they are available for congestion estimation use, but availability is very limited at this moment.

Meanwhile, mobile phone-based crowd estimation and people counting is a recent hot topic due to its low-cost feature. People density and mobility estimation in urban areas has been investigated so far. Ref.[16] presents a method to model human mobility in large-scale urban areas by utilizing cellular tower information, which can be obtained without sensing or proximity detection between mobile phones. On the other hand, for indoor, narrower regions, Ref.[20] presents a unique approach that obtains the number of mobile devices existing in an area based on audio tones. More recently, some studies show the possibilities of flock and crowd detection using inertial sensors and communication devices of mobile phones. Refs.[22, 23] use multi-modal sensors in mobile phones and WiFi signals for flock detection, and Ref.[35] uses mobile phones for estimating crowd density, counting the number of devices and measuring RSSI fluctuation, leveraging collaboration via Bluetooth between users in proximity. These approaches are beneficial in the sense that density estimation or detection of crowd can be achieved by off-the-shelf mobile phones and simple infrastructures such as WiFi. By contrast, our target is car-level congestion estimation in a train environment which is different from the approaches above. In such an environment, we need not only congestion estimation but also car-level position estimation since congestion levels are different for each car and car-level positions are difficult to obtain. Moreover, we cannot expect a large number of Bluetooth devices since the number of passengers per car is much less compared to large events such as festivals.

**Mobile Phone Localization**

Since mobile phone localization has been well-studied, there have been a variety of approaches so far. However, localization on trains is completely different from such localization since signal attenuation often occurs due to human bodies and doors between cars as well as distance. Therefore, In this viewpoint, general WiFi-based techniques [36, 17] are not applicable to position and congestion estimation in cars. We may expect support from GPS signals as supposed in Ref.[5], but GPS is not essentially sufficient to identify car-level positions. Position estimation on station platforms before boarding may be helpful to estimate car-level positions. However, we may face with the same problem with train cases – we need to take into account both congestion situation and location, and may become more difficult than train cases due to higher mobility of passengers on platforms. A possible alternative is pedestrian dead reckoning (PDR) such as Refs.[2, 10, 18, 24], which has been studied for location-based services in ubiquitous computing. More recently, mobile phone-based PDR as Refs.[18, 13, 26] has been investigated to enhance applicability for various services. PDR techniques have been sophisticated, nevertheless large error still occurs as the moving distance is longer due to irregular motions. Therefore, in order to minimize the error, Ref.[34] uses the combination of PDR and map matching which uses the characteristic change of data on geomagnetic sensors and accelerometers. However, collision avoidance motion is not considered though it is often seen on station platforms.

**Contributions**

Contributions of this paper are summarized as follows. Firstly, the target problem itself is unique since, as far as we know, there has been no approach that aims at sensing and estimating those passengers’ “live” positions and congestion information. Secondly, we provide a reasonable and highly-accurate approach to the research goal. The proposed method is simple and robust since it only relies on Bluetooth as well as communication capability with a server (e.g. 3G/LTE). No dedicated devices nor computation are necessary on the phones (the only thing they need to do is scanning neighboring Bluetooth nodes and sending its summary to the server). We provide a novel algorithm of position and congestion estimation as a Bayesian likelihood updater.

The technique of time-sequential and simultaneous update of two correlated parameters is well exploited for robotic SLAM [9, 6, 15], but they focus on location estimation of moving objects and its static surroundings. On the other hand, the proposed method considers train-specific features where (i) Bluetooth signal propagation is affected by distance, crowd of passengers, and the structure of trains, (ii) most passengers on board do not usually change their positions, and (iii) not all the passengers get off the train at a station. Finally, we have collected over 50,000 Bluetooth samples in real trains
of different lines in Osaka for more practical design of the method and performance in the real world.

DATA ANALYSIS FROM PRELIMINARY EXPERIMENT

Received Signal Strength Indication (RSSI) is often used to estimate distances between wireless devices. However, the path-loss model, which is often used for distance estimation, may not work well due to train-specific features such as signal attenuation by doors between neighboring cars (if exist) and human bodies in case of congestion. Furthermore, reflections by metal walls, ceilings and floors affect radio propagation, which may produce unpredictable results. Therefore, in order to observe how signals propagate in cars, we have conducted preliminary experiments.

Data Collection Environment

We have collected more than 50,000 Bluetooth samples over 120 minutes on Midosuji line, which is one of the major railway lines in Japan. 17 participants stood in cars as shown in Fig.1 with mobile phones. We let each mobile phone carry out Bluetooth inquiry scan (for 12 seconds) and inquiry transmission alternately. We used Nexus S with Android version 4.1 for the experiments. There are 23 stations on Midosuji line. The length of the car is 18.74m and the seating capacity of 44 people per car.

Analysis of Bluetooth RSSI

Figure 2 shows the distance versus the average RSSI. We note that the average RSSI is calculated from RSSI observations during $T$ seconds for mitigating noise effects. We use $T = 60$ in the following experiments. The trend of signal attenuation by distance can somewhat be seen although the standard deviations of RSSI are large especially at distance shorter than 18m (the length of a car). This means distance estimation based on RSSI may not work properly. It is also observed that RSSI does not show any large difference at the distance of car length or longer since the effect of doors between cars is a significant factor. In summary, estimating the exact distance from such fluctuated RSSI may not be a good idea.

To see the capability of relative position and congestion estimation by RSSI, we have conducted further analysis on the characteristics. Because the effect of doors between cars is dominant from the above observation, we classify node pairs by the following three types based on their relative positions in cars: a pair $(i, j)$ of nodes is called **immediate neighbor pair** if $i$ and $j$ are in the same car, or **adjacent neighbor pair** if $j$ is in the $i$’s adjacent car and vice versa. Otherwise it is called **separate node pair**. In addition, considering the seating capacity of 44 people per car and our experience, a car is said to be **crowded** if there are more than 60 passengers in the car, and **uncrowded** otherwise. We employ these two levels to represent congestion of cars.

Figure 3 shows the distance versus average RSSI for each category described above. Data used in this figure is the same as Fig.2. Moreover, it is defined as **adjacent neighbor pair & crowded** if at least one of two cars is crowded. Otherwise it is defined as **adjacent neighbor pair & uncrowded**. At first, two nodes of a separate node pair were not able to detect each other, and there were only 100 samples for such pairs out of 50,000 (about 0.002%). Moreover, for separate node pair in Fig.3, their RSSI was very weak, and the values were close to the sensitivity threshold of RSSI (about -90dBm). This is because door(s) between cars has a great influence on attenuation. Therefore, we conclude that if RSSI is observed between a pair, it is substantially considered as an immediate neighbor pair or an adjacent neighbor pair. Fig.3 shows that when strong RSSI ($>-70$dBm) is detected, the pair is an immediate neighbor pair with a high probability since most of RSSI stronger than -70dBm are observed by immediate neighbor pairs with short distance only. On the other hand, in the case that RSSI is less than -70dBm, it seems difficult to distinguish an immediate neighbor pair and an adjacent neighbor pair. In order to classify node pairs in such cases, we need additional information, i.e. relationships of pair categories with other nodes. For example, if $(i, j)$ and $(j, k)$ are immediate neighbor pairs, $(i, k)$ is also an immediate neighbor pair with a high probability.

As for the effect of congestion, Fig. 3 shows that the average RSSI is partly different between crowded and uncrowded due to the influence of human bodies. However, the difference is not large compared to the difference seen in the classification of pairs. This result indicates that signal attenuation due to doors between cars is much larger than that by human bodies.
The above analysis indicates that RSSI may distinguish immediate neighbor pairs and adjacent neighbor pairs regardless of congestion levels. To clarify the classification ability by RSSI, we show the average RSSI distributions of the two pair categories in Fig. 4. In the distributions, the congestion levels and the intermodal distances are ignored. From Fig. 4, we can see that two distributions do not overlap in the strong RSSI part. This indicates that we can classify node pairs if the observed RSSI is strong enough. However, the classification may not be reliable when the observed RSSI is weak. As mentioned earlier, in order to accurately determine the relative positions of node pairs from the observed RSSI, it is necessary to use relationships of pair categories with other nodes. In addition, we may be able to identify node pairs by repeating RSSI observations over time since the distributions of two categories in Fig. 4 are different.

From the above analysis, we conclude that we can estimate the car-level relative positions of node pairs regardless of congestion levels. It is also found that congestion levels can be estimated if we focus on RSSI of immediate neighbor pairs in each car. Therefore, we design a method to estimate car-level positions followed by congestion estimation. The proposed method estimates the categories of node pairs from RSSI and relationships of pair categories with other nodes. Furthermore, considering train specific mobility features where most passengers on-board do not change their cars, we maintain the likelihoods over time and estimate their relative positions.

It is worth noting that we could observe 3-7 anonymous devices (non-users) during each period between stations. This is one of advantages of using Bluetooth since RSSI from such non-users helps increase the amount of observations. We note that it is known that there are some differences in RSSI depending on devices [14]. However, the heterogeneity of mobile phones does not have large effects because the effect of doors between cars is larger than the RSSI difference due to the device heterogeneity.

CAR-LEVEL CONGESTION AND POSITION ESTIMATION

System Architecture

The proposed method is composed of mobile devices (called nodes) with Bluetooth and a server for analyzing the RSSI observed between them. Bluetooth observations are aggregated from mobile devices via cellular networks. In the proposed method, trip time of a train in each inter-station consists of observation and estimation phases. The trip time in an inter-station is called a section hereafter. An observation phase is followed by an estimation phase. It starts at the time of departure from a station and ends after T seconds. In the observation phase of section s, each user i scans its neighbors’ inquiry messages and also transmits its inquiry to its neighbors. Once it receives an inquiry from a Bluetooth device j (j may be a user or non-user), i records the Bluetooth IDs of i and j with the RSSI value. i iterates this scan for T seconds and calculates the average RSSI $r_{ij}$, which is later sent to a server via cellular networks by the end of the observation phase. In the subsequent estimation phase, the server first estimates relative positions of node pairs in the train using the gathered average RSSI. Then, the proposed method categorizes all of node pairs into immediate neighbor pairs, adjacent neighbor pairs, or separate node pairs. From the analysis in the preliminary experiment, separate node pairs are identified when no RSSI is observed. For categorization of the other pairs, we use the average RSSI and the RSSI model learned beforehand. From the above categorization, we know car-level relative positions between nodes.

In order to estimate car-level absolute positions (i.e. car IDs), we assume a few nodes called reference nodes whose car IDs estimation results are considered more reliable than the others. Such reference nodes may be train conductor and crew, passengers staying trains longer than the others, or some limited incentive users with or without reward to input their exact car IDs. We note that reference nodes are not always necessary. This is because once car IDs of some nodes are estimated with high reliability, such nodes can be regarded as reference nodes. Car IDs of nodes are estimated based on car IDs of reference nodes and car-level relative positions between nodes. Finally, the congestion level of each car is estimated from the average RSSI among nodes which exist in the same car.

We assume that departures of trains (start time of the observation phases) can be recognized by inertial sensors of mobile phones such as accelerometers so that they can start Bluetooth scans automatically, and we can identify a set of nodes existing in the same train. This can easily be achieved by train timetables and rough location information obtained from GPS, WiFi, and so on. We also assume that a train consists of a sequence of $|V|$ cars where $V = \{v_1, v_2, \ldots, v_{|V|}\}$ and $v_i$ and $v_{i+1}$ are directly connected (adjacent). Hereafter, we explain the estimation for nodes boarding on the same train.

Overview of Observation Phase

When both i and j are users participating in the system, the average RSSI $r_{ij}$ and $r_{ji}$ are transmitted to the server. From the observations of both sides, we define the unique average RSSI between i and j. Weak RSSI can be observed at short distances due to the effect of shadowing by human bodies while strong signals are rarely observed at long distances. This indicates strong signals are more reliable than weak signals. Therefore, we define the average RSSI between i and j as $r_{ij} = r_{ji} = \max(r_{ij}, r_{ji})$. We note that the proposed method utilizes observations $r_{ij}$ even if non-user j is observed by

1. Each scan duration is about 12 seconds in our implementation.
2. Section s and/or nodes i and j may be omitted for simplicity of notations according to the context.
user $i$. This helps increase the accuracy by increasing the amount of observations. We note that the privacy concern about revealing unique IDs to others and the server is small because the proposed method needs short-term ID tracking only. For example, each user can anonymize its own ID every time when he boards a train.

### Overview of Estimation Phase

In the estimation phase, the proposed method estimates node locations and car congestion based on the observations collected from users. The server maintains car likelihood $l_i^s(v_k)$ which indicates the likelihood that node $i$ exists in car $v_k$ in section $s$. In the position estimation, the observation $r_{ij}^s$ is mapped to the probability that $(i, j)$ is an immediate neighbor pair, based on the Bluetooth RSSI model learned beforehand. This probability is called an immediate neighbor probability, and represents the relative location between a pair of nodes. Then, the car likelihood of each node is updated. The basic idea is to propagate car-level position information of reference nodes according to the immediate neighbor probabilities. For example, let $j$ be a reference node in car $v_k$. If $r_{ij}$ is strong enough to believe $(i, j)$ is an immediate neighbor pair, $l_i^s(v_k)$ is considered to be high. On the other hand, if $r_{ij}$ is weak, $l_i(v_{k-1})$ and $l_i(v_{k+1})$ should be high since $(i, j)$ may be an adjacent neighbor pair.

To increase the accuracy of position estimation, the proposed method applies Bayesian inference to the update of immediate neighbor probabilities and car likelihoods in section $s$ by using those in the previous section $s-1$ under the assumption that most passengers do not move to other cars during their trips. Also, we use a negative observation, which is the fact that no RSSI is observed between nodes. From the negative observation between $i$ and $j$, it can be inferred that $(i, j)$ is not an immediate neighbor pair.

Finally, the proposed method estimates the congestion level of each car by combining the estimated car likelihoods and congestion likelihoods. The congestion likelihood between $i$ and $j$ is also derived by the average RSSI $r_{ij}$, based on the learned Bluetooth RSSI model. The congestion level of car $v_k$ is estimated by taking a kind of majority decision of congestion likelihoods weighted by car likelihoods.

### Car-Level Position Estimation

#### Assignment of Immediate Neighbor Probability

Let $S_{ij}^s$ be the event that a node pair $(i, j)$ is an immediate neighbor pair in section $s$. The immediate neighbor probability $P(S_{ij}^s|r_{ij}^s)$ is defined as below. From Bayes' theorem, the posterior probability that $(i, j)$ is an immediate neighbor pair given $r_{ij}^s$ is

$$P(S_{ij}^s|r_{ij}^s) = \frac{P(r_{ij}^s|S_{ij}^s)P(S_{ij}^s)}{P(r_{ij}^s|S_{ij}^s)P(S_{ij}^s) + P(r_{ij}^s|S_{ij}^s)P(S_{ij}^s)} \quad (1)$$

where the likelihood functions $P(r_{ij}^s|S_{ij}^s)$ and $P(r_{ij}^s|S_{ij}^s)$ are given from the Bluetooth RSSI model learned beforehand. Since the proposed method assumes nodes do not move to other cars during their trips, the prior probability $P(S_{ij}^s)$ is $P(S_{ij}^s) = P(S_{ij}^{s-1}|r_{ij}^{s-1})$.

### Algorithm 1 Updating Car Likelihoods in Section $s$

**Require:** Node Set $N^s$, Reference Node Set $RN^s(\subseteq N^s)$, and Car Sequence $V$

**Ensure:** $\{l_i^s(v_k)|v_k \in N^s, \forall v_k \in V\}$

1. **for** $i \in N^s, v_k \in V$ **do**
   1. **Initialize** $l_i^s(v_k)$ by Eq.2 or Eq.3
2. **end for**

3. **/* Calculate immediate neighbor probabilities */**
   1. **for** $i, j \in N^s$ **do**
      1. Calculate $P(S_{ij}^s|r_{ij}^s)$ by Eq.1
   2. **end for**

4. **/* Update car likelihoods */**
   1. **for** $t = 1$ to $R$ **do**
      1. **for** $i \in N^s - RN^s, v_k \in V$ **do**
         1. Update $l_i^s(v_k)$ by Eq.4
      2. **end for**
   2. **for** $i \in V - RN^s$ **do**
      1. $l_i^s(v_k) \leftarrow l_i^s(v_k)/\sum_{v_k \in V} l_i^s(v_k)$ **/* Normalization */**
   3. **end for**

We note that if at least one of $i$ and $j$ is a new passenger, $1/|V|$ is assigned to the prior probability $P(S_{ij}^s)$ since we do not have any knowledge.

#### Updating Car Likelihood

We describe the algorithm to update car likelihoods in section $s$ in Algorithm 1. Let $N^s$ be a set of users and non-users in section $s$. We also denote a set of reference nodes in section $s$ as $RN^s$. In the initialization, $l_i^s(v_k)$ is set to $l_i^{s-1}(v_k)$ if $j$ was on board in the previous section $s-1$. When $j$ is a new passenger, $l_i^s(v_k) = 1/|V|$ (Eq.2). When $i$ is a reference node, $l_i^s(v_k) = 1$ if $i$ exists in car $v_k$ and the other car likelihoods of $i$ are set to 0 (Eq.3).

$$i \in N^s - RN^s \Rightarrow l_i^s(v_k) = \begin{cases} l_i^{s-1}(v_k) & \text{if } i \in N^{s-1} \\ 1/|V| & \text{otherwise} \end{cases} \quad (2)$$

$$i \in RN^s \Rightarrow l_i^s(v_k) = \begin{cases} 1 & \text{if } i \text{ exists in car } v_k \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$l_i^s(v_k)$ is updated based on car likelihoods of all nodes and car-level relative positions estimated by the average RSSI related to $i$. If $(i, j)$ is an adjacent neighbor pair, $l_i^s(v_k)$ and $l_j^s(v_k)$ should not be high at the same time since $i$ and $j$ probably do not exist in the same car $v_k$. In the other cases, $l_i(v_k)$ is updated according to $l_i^s(v_k)$ and its immediate neighbor probability $P(S_{ij}^s|r_{ij}^s)$. From the definition of $P(S_{ij}^s|r_{ij}^s)$, the effect of $l_i(v_k)$ should be strong when $P(S_{ij}^s|r_{ij}^s)$ is high. On the other hand, $i$ exists in car $v_k$ or $v_{k+1}$ when $j$ exists in car $v_k$ and $P(S_{ij}^s|r_{ij}^s)$ is low. Therefore, we have designed the algorithm so as to update $l_i(v_k)$ by calculating the sum of $l_j(v_{k-1}), l_j(v_{k+1})$, and $l_j(v_k)$ weighted by $P(S_{ij}^s|r_{ij}^s)$. The algorithm iterates the update process $R$ times in order to propagate car likelihoods of reference nodes to all nodes.

The update of car likelihoods is as follows. Let $N^s_0$ be a set of nodes $j$ whose average RSSI $r_{ij}$ is available. Then, a negative observation set $N^s - N^s_0$ includes all the other nodes
whose RSSI are not observed, i.e. neither \(i\) nor \(j\) observes any RSSI from \(j\) or \(i\). We also denote a set of adjacent car(s) of \(v_k\) as \(W_k \subset V\) whose size is \(|W_k|\). To clarify the round of iteration, let \(l_i^{s,t}(v_k)\) denote the \(t\)-th round result of the car likelihood of node \(j\) at car \(v_k\) in section \(s\). The \(t+1\)-th update of \(l_i^{s,t+1}(v_k)\) is as below.

\[
l_i^{s,t+1}(v_k) = \sum_{j \in N^r_i} \left[ l_j^{s,t}(v_k) P(S^r_i | r_{ij}) \right] + \frac{1 - P(S^r_i | r_{ij})}{|W_k|} \sum_{w \in W_k} l_s^{w,t}(w) \cdot \text{negative}_i^{s,t}(v_k)
\]

(4)

where the function \(\text{negative}\) is defined for the correction by the negative observation as below.

\[
\text{negative}_i^{s,t}(v_k) = \sum_{j \in N^r_i} \left[ \frac{1 - l_j^{s,t}(v_k)}{1 - l_j^{s,t}(v)} \right]
\]

(5)

The function \(\text{negative}\) helps increase the accuracy of position estimation. For example, let \(j\) be a node in car \(v_k\). If \(P(S^r_i | r_{ij})\) is low, it is considered that \(j\) exists in the neighboring car of \(v_k\) (i.e. \(v_{k-1}\) or \(v_{k+1}\)). However, because we cannot determine whether \(i\) exists in \(v_{k-1}\) or \(v_{k+1}\), \(i\)'s car likelihoods for the both cars (i.e. \(l_i^{s,t}(v_{k-1})\) and \(l_i^{s,t}(v_{k+1})\)) are equally increased according to \(l_j(v_k)\) and \(P(S^r_i | r_{ij})\). This causes the problem of a geometric flip on the car-level. By applying the function \(\text{negative}\), we can suppress the effect of geometric flips by reducing \(l_i^{s,t}(v_{k-1})\) or \(l_i^{s,t}(v_{k+1})\) when many separate nodes of \(i\) exist in \(v_{k-1}\) or \(v_{k+1}\).

In the end of each round, the car likelihood of each node is normalized and used in the next round. For the accurate estimation, the proposed method regards nodes whose highest car likelihoods are greater than \(l_{TH}^C\) as new reference nodes (although they are somewhat ambiguous) when reference nodes with manual input do not exist. The car likelihoods of them are fixed and never updated for convergence of the results until more reliable reference nodes get on.

**Car Congestion Estimation**

The congestion level of each car is estimated by using the estimated car likelihoods and the average RSSI among nodes. Let \(C_i^{s,t}\) denote the event that the space between nodes \(i\) and \(j\) is crowded in section \(s\). From Bayes’ theorem, the posterior probabilities that the space between \(i\) and \(j\) is crowded and uncrowded are

\[
P(C_i^{s,t} | r_{ij}) \propto P(r_{ij} | C_i^{s,t}) P(C_i^{s,t})
\]

\[
P(C_i^{s,t} | r_{ij}) \propto P(r_{ij} | C_i^{s,t}) P(C_i^{s,t}).
\]

(6)

The definition of congestion levels depends on environments and requirements. In this paper, we define the crowded environment is the situation where no seat is available since such information is useful for many passengers.

The congestion level of car \(v\) is estimated by taking a likelihood ratio of the weighted sums of crowded likelihoods \(P(C_i^{s,t} | r_{ij})\) and uncrowded likelihoods \(P(C_i^{s,t} | r_{ij})\). For the estimation of congestion in car \(v\), we consider node \(i\) exists in car \(v\) if \(l_i^{s,t}(v)\) is the highest. We let \(N^v(v)\) denote a set of nodes whose car likelihoods of \(v\) are the highest. For each pair of nodes \(i, j \in N^v(v)\), the proposed method uses the same weight \(l_i^s(v) \cdot l_j^s(v)\) to crowded and uncrowded likelihoods of \(v\) in section \(s\). This is natural because the weights indicate likelihoods that the estimated probabilities of congestion arise from the observations in the car. The congestion likelihood ratio \(l^{s,t}(v)\) of \(v\) in section \(s\) is defined as below.

\[
l^{s,t}(v) = \sum_{i,j \in N^v(v)} l_i^s(v) l_j^s(v) P(C_i^{s,t} | r_{ij}) P(C_j^{s,t}) P(C_i^{s,t})^{-1}
\]

In general, since the congestion changes for each section, it is not appropriate to reflect the congestion likelihoods of the previous section in the calculation of the current section. Therefore, the prior probabilities \(P(C_i^{s,t})\) and \(P(C_j^{s,t})\) are generally unknown and thus set to 1/2. However, previous knowledge such as rush hours can be given if available. The congestion level is determined based on crowded and uncrowded thresholds \(C_{TH}^C\) and \(C_{TH}^U\). If \(l^{s,t}(v) > C_{TH}^C\), \(v\) is estimated as crowded. Similarly, if \(l^{s,t}(v) < C_{TH}^U\), \(v\) is estimated as uncrowded.

**Learning Bluetooth RSSI Model**

Our method assumes the Bluetooth RSSI models, i.e. the likelihood functions \(P(x | S), P(x | S)\) \(P(r | C)\) and \(P(r | C)\) learned beforehand. It may be effective to learn the model for each railway line since characteristics of radio propagation may be different depending on materials and structure of cars. We investigate the effect of different railway lines in the experiment later. In this section, we describe the Bluetooth RSSI models learned for Midosuji line.

In the learning phase, we have collected RSSI for 1 hour using 16 mobile phones in each event \(S, S, C\) and \(C\). The participants were uniformly located as shown in Fig.6(a). The RSSI in \(T\) seconds after departure of the train from each station are averaged and used for the learning. The Gaussian fitting was applied to the distributions of average RSSI for each event. We used maximum likelihood estimation for its fitting parameters. Let \(f(r, \mu, \sigma)\) denote the probability density function of the normal distribution with the mean \(\mu\) and the standard deviation \(\sigma\). Let also denote \(\mu_E\) and \(\sigma_E\) the mean and the standard deviation of the fitted Gaussian distribution of an event \(E\), respectively. Then, we define the likelihood function of \(E\) as

\[
P(r | E) = f(r, \mu_E, \sigma_E)
\]

\[
= \int_{-dh}^{dh} \frac{1}{\sqrt{2\pi \sigma_E^2}} \exp\left(-\frac{(r - \mu_E)^2}{2\sigma_E^2}\right) dr.
\]

(7)

Fig.5 shows the likelihood functions of each event with its mean and standard deviation for Midosuji line.

**PERFORMANCE EVALUATION**

**Environment and Parameter Settings**

We have evaluated the proposed method through real experiments where Bluetooth samples were collected over 259 minutes in 4 railway lines: Midosuji (M), Osaka Loop (OL), Hankyu Senri (H), and Osaka Monorail (OM). Table 1 shows the specifications of cars in the lines. The likelihood functions for neighbor pair categories and congestion levels were built based on the Bluetooth samples collected in the preliminary
experiment conducted on Midosuji line. We note that the data set used in the following evaluation is collected on another day, i.e. the data set under test is completely independent of those for building likelihood functions.

There are two types of scenarios, Uniform and Random, for M, OL, and H lines. The number of participants was 16 in both scenarios. The positions of participants in the Uniform scenario are shown in Fig. 6(a), which are uniformly distributed over 4 cars. Each participant stood at the designated position hanging Nexus S from his neck. In the Random scenario, each participant was asked to board the designated car without any other limitations to consider natural situations (see Fig. 6(b)). Therefore, some were seated while some others stood at somewhere in the cars. For OM line, we have used only the Uniform scenario in order to see the performance in the cars without doors between them. Hence, the number of participants was 6 in OM scenario. Throughout the experiments, each mobile phone had carried out Bluetooth inquiry scan for 12 seconds and inquiry transmission alternatively. We define $C$ as the event that there are more than 60 ppl/car based on the seating capacities in Tab. 1. The snapshots of congestion levels are shown in Fig. 7. We used $R = 10$, $T = 60$ sec., and $b_{TH}^C = b_{TH}^C = 1.0$ as the default settings. For reference nodes, we selected nodes 5 and 12 in M, OL, and H lines and node 6 in OM line. The car IDs of reference nodes were given manually. For the evaluation, we introduce a metric called top-k accuracy to see the accuracy of car-level position estimation in detail. We call an estimation result (i.e. a car likelihood) of a node is k-correct if the likelihood of the car where the node exists is the k-th highest or above. Then, the top-k accuracy is defined as

$$\frac{\text{# of k-correct nodes}}{\text{# of nodes}}.$$  

We simply denote the top-1 accuracy as position accuracy in the following sections.

### Accuracy of Immediate Neighbor Probability

To see the ability of immediate neighbor probabilities to distinguish between immediate neighbor pairs and adjacent neighbor pairs, Fig. 8 shows the cumulative distribution functions of immediate neighbor probabilities of adjacent pairs, immediate pairs, and immediate pairs without continuous estimation. In the case of the immediate pairs without continuous estimation, immediate neighbor probabilities in the previous section are not used for the prior probabilities $P(S_{ij})$ in Eq. 1. Instead, $1/4$ is assigned. From the result, we can see that the probabilities of almost 90% of immediate neighbor pairs are nearly 1.0 while those of 90% of adjacent neighbor pairs are less than 0.4. This indicates that the proposed method is able to distinguish between immediate neighbor pairs and adjacent neighbor pairs accurately. Fig. 8 also shows that the probabilities without continuous estimation are low compared to the proposed method. Hence, we could confirm the effectiveness of the long-term estimation.

### Accuracy of Position Estimation

#### Effects of Parameters and Environments

To see the effects of parameters and environments on position estimation, we have evaluated the position accuracy of all nodes in different settings under Uniform scenario of M line.

Figure 9(a) shows the update rounds $R$ versus the accuracy. The two lines in this figure show the accuracy of the first section and the average accuracy of all the sections. From the result, we can see that it requires 5 rounds to converge in the first section. This is because car likelihoods of reference nodes propagate one hop per round. Therefore, the proposed method needs several rounds to propagate reference nodes’ car likelihoods to all the nodes sufficiently. On the other hand, in the case of the average of all the sections, $R = 2$ is enough to converge because car likelihoods of reference nodes have

### Table 1: Railway Lines and Car Specifications

<table>
<thead>
<tr>
<th>Railway Line</th>
<th>Stations</th>
<th>Cars</th>
<th>Car Length</th>
<th>Car Width</th>
<th>Car Height</th>
<th>Seat Cap.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midosuji (M)</td>
<td>23</td>
<td>10</td>
<td>18.74</td>
<td>2.89</td>
<td>18.74</td>
<td>3.74</td>
</tr>
<tr>
<td>Osaka Loop (OL)</td>
<td>20</td>
<td>8</td>
<td>19.50</td>
<td>2.80</td>
<td>19.50</td>
<td>4.14</td>
</tr>
<tr>
<td>Hankyu Senri (H)</td>
<td>12</td>
<td>7</td>
<td>18.90</td>
<td>2.80</td>
<td>18.90</td>
<td>4.09</td>
</tr>
<tr>
<td>Osaka Monorail (OM)</td>
<td>6</td>
<td>4</td>
<td>14.60</td>
<td>2.98</td>
<td>14.60</td>
<td>5.20</td>
</tr>
</tbody>
</table>

We used $R = 10$, $T = 60$ sec., and $b_{TH}^C = b_{TH}^C = 1.0$ as the default settings. For reference nodes, we selected nodes 5 and 12 in M, OL, and H lines and node 6 in OM line. The car IDs of reference nodes were given manually. For the evaluation, we introduce a metric called top-k accuracy to see the accuracy of car-level position estimation in detail. We call an estimation result (i.e. a car likelihood) of a node is k-correct if the likelihood of the car where the node exists is the k-th highest or above. Then, the top-k accuracy is defined as

$$\frac{\text{# of k-correct nodes}}{\text{# of nodes}}.$$  

We simply denote the top-1 accuracy as position accuracy in the following sections.
fig 8: CDFs of Immediate Neighbor Probabilities

Fig 9: Average Accuracy vs. Parameters

been already propagated to many nodes in the update of the previous section. Consequently, the update rounds \( R \) should be set to 5 or above. This is because the case of the first section is regarded as the worst case since all the nodes except reference nodes do not have any knowledge about their car positions. Fig.9(b) shows the observation duration \( T \) versus the average accuracy of all the sections. It is observed that the accuracy increases with the increase of the observation duration. This is simply explained by the result that more amount of observations leads to noise mitigation. Fig.9(c) shows the effect of the number of participants. In this evaluation, we have simulated the intended number of participants from the real data. For example, when the number of participants per car is one, we have chosen one participant from 4 participants in each car and removed observations related to the other nodes from evaluation. We note that reference nodes 5 and 12 are always chosen. Then, we have averaged the accuracy of all the cases and all the sections. The result of Fig.9(c) shows that the accuracy increases as the number of participants increases. This is because the amount of the relative location information increases when there are many participants. The number of reference nodes largely affects the accuracy as we can see from Fig.9(d). We note that similarly to the evaluation of the number of participants, we have simulated the test cases from the real data and plot the average of all the sections and all the cases that satisfy the intended numbers of reference nodes. From the result, the improvement from 1 to 2 reference nodes is especially significant.

This is because if the number of reference nodes is only one, there is a possibility of a geometric flip. Overall, the accuracy increases as the number of reference nodes increases, which is natural.

**Optimization of RSSI Model**

We have evaluated the effect of different models (i.e. likelihood functions) since structures of cars are different depending on railway lines and they may affect the accuracy. For the evaluation, we prepared a self-fitted model and compared its accuracy with M line model. Each self-fitted model is built by using the data set identical to the test data set and used as the upper bound of the accuracy. As we can see from the results in Tab.2, the accuracy increases when the self-fitted models are used for all the railway lines. However, it is also remarkable that the accuracy using M line model for OL and H lines is comparable to that using self-fitted models. The reason is that the car structures of those three lines are like each other. In contrast, the accuracy improved by 10% if we use the self-fitted model for OM line since there is no door between cars of OM line, which is clearly different from other three lines. The above results indicate that it is effective to use appropriate models for each car structure.

**Effects of Geometric Distribution of Reference Nodes**

In the proposed method, it is considered that the relationship of relative positions between reference nodes affects the accuracy. In order to see the effect of geometric distribution of reference nodes, we show the accuracy for different reference node deployments in Tab.3. If the number of reference nodes is one, the accuracy is 0.651 because geometric flips occur. Even when the number of reference nodes is two, the accuracy is only 0.74 if it is an immediate neighbor pair because those two reference nodes propagate almost the same car likelihoods, which do not help increase the car likelihoods of others and may cause geometric flips. On the other hand, the accuracy is more than 0.90 when two reference nodes are distributed in different cars. The only exception is the case of 3 cars away, which is 0.875. This is because car likelihoods from reference nodes deteriorates and become uncertain hop by hop.

**Sustainability for Loss of Reference Nodes**

<table>
<thead>
<tr>
<th>Railway Line</th>
<th>Likelihood Function</th>
<th>M</th>
<th>OL</th>
<th>H</th>
<th>OM</th>
</tr>
</thead>
<tbody>
<tr>
<td>M line Model</td>
<td></td>
<td>0.957</td>
<td>0.804</td>
<td>0.975</td>
<td>0.791</td>
</tr>
<tr>
<td>Self-fitted Model</td>
<td></td>
<td>-</td>
<td>0.814</td>
<td>0.990</td>
<td>0.875</td>
</tr>
</tbody>
</table>

Table 2: Effect of Car Likelihood Function (Average accuracy of all sections is shown)

<table>
<thead>
<tr>
<th># of RNs</th>
<th>Category of Reference Node Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.651 0.929 0.928 0.875</td>
</tr>
</tbody>
</table>

Table 3: Effect of Reference Node Deployment
To see the effect of new reference nodes selected after default reference nodes exit the train, we have simulated the scenario where all default reference nodes exit the train at the end of section 4. Fig. 10 shows the accuracy for each section. We have empirically set the threshold $l_{TH}$ for new reference nodes to 0.4. In the case without selecting new reference nodes, the accuracy quickly degrades after section 5 since no reference node exists. On the other hand, it is clear that the degradation of the accuracy is mitigated by new reference nodes. However, some degradation in the middle sections is also observed compared to the case where default reference nodes do not exit the train. This cannot be avoided since car likelihoods of new reference nodes are less certain than those of default ones. This effect is significant in the middle sections where many passengers are on board since RSSI tends to be weak due to human bodies, which leads to deterioration of car likelihoods over distance. However, in the later sections, the accuracy improves since the number of passengers decreases. Moreover, if many passengers are on board, we can sufficiently expect that the number of participating users increases and new reference nodes get on. Therefore, the degradation of the accuracy as shown in this figure may be reduced. From the results above, we could confirm the sustainability of our method for loss of reference nodes.

Accuracy of Congestion Estimation

We have evaluated the performance of congestion estimation in M line. Fig. 11 shows the confusion matrices of 4 cases: (a) Uniform scenario, (b) Uniform scenario with ideal car likelihoods, (c) Random scenario, and (d) Random scenario with ideal car likelihoods. The number of sections in M line is 22. The numbers of cars in Uniform scenario and Random scenario are 4 and 3, respectively. Hence, the estimation times in Uniform scenario and Random scenario are $22 \times 4 = 88$ and $22 \times 3 = 66$, respectively. In the scenarios with ideal car likelihoods, all nodes are assumed that the proposed method has already known their boarding cars as baselines for comparison. In this evaluation, we use precision, recall, and F-measure for metrics. F-measure is used to see the trade-off between these two metrics as below.

$$F\text{-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Fig. 11 shows that the estimation results are similar to their ideal cases in both scenarios. The F-measures of the estimation results are slightly worse than the ideal cases. It is also noticeable that F-measures of crowded events in Uniform scenario are low. This is caused by uniform distance among nodes where there is no observation at short distance. In fact, the F-measures of crowded events in Random scenario are greater than 0.8. We can rarely obtain RSSI strong enough to believe the space is uncrowded in Uniform scenario in uncrowded events because the node deployment of Uniform scenario is sparse compared to that of Random scenario. In real use cases, we can expect that the number of participants increases with the increase of the congestion level and their positions in a car are randomly distributed. Therefore, from the result in Random scenario, it is confirmed that the proposed method can achieve F-measure of 0.8 on average.

The above results in Fig. 11 indicate uncrowded events in Uniform scenario are often wrongly estimated as crowded events. The proposed method may be able to avoid such wrong estimation of uncrowded events by adjusting the threshold $l_{TH}^C$ for congestion likelihoods to a conservative (high) value. Fig. 12 shows recall and precision of crowded events in Uniform scenario with ideal car likelihoods. As we can see from the result, precision increases with the increase of the threshold while recall degrades. The setting of the threshold is application-dependent, thus we have shown the ability to adjust the congestion estimation accuracy in terms of precision and recall.

Moreover, we can apply prior knowledge on congestion levels if available. For example, the line in Fig. 13 shows the number of passengers at each station in M line according to Ref.[8]. From the data, we set the prior probability $P(C)$ of the congested event as shown in Fig. 13. The results are shown in Fig. 14. It is obvious that the accuracy of the congestion estimation can be increased if prior knowledge is available. Such prior knowledge can also be obtained by smart cards as described in [25].

Accuracy in Real Scenario

Finally, we have evaluated our method in a real scenario where passengers exit/board the train at each station. For the evaluation, we used Random scenario to simulate a real
scenario. In the real scenario, we reproduced natural behavior of passengers: the number of participants increases at stations before main stations and many participants exit the train at the main stations. We randomly chose 4 default reference nodes beforehand from all participants. $T_H = 0.4$ was used to choose new reference nodes when no reference node exists. Fig. 15 shows the numbers of participants, reference nodes, and participants boarding/exiting the train at each station. The accuracy is also shown by lines in Fig. 15. The result shows that our method could achieve the accuracy of 0.83 on average. The top-2 accuracy also achieved 0.96, which is highly accurate and still useful as position information. We note that no reference node exists in some sections. Even in such sections, our method keeps the high accuracy due to selecting new reference nodes in current participants. For the performance of the congestion estimation, the average F-measure was 0.75 where those of uncrowded and crowded events were 0.79 and 0.71, respectively. With prior knowledge as shown Fig. 13, the average F-measure was improved to 0.82. From the above results, we could confirm the proposed method achieves estimation of positions and congestion accurately in real environments.

**PROTOTYPE IMPLEMENTATION**

We have built an Android prototype of a navigation system based on the proposed method. Passengers on board can see their car likelihoods by color bars as shown in Fig. 16(a). According to the estimated car likelihoods, the system displays optimal routes including elevators, stairs, and exits from the estimated cars to their destinations. Meanwhile, passengers planning to board trains can see the current congestion level of each car as shown in Fig. 16(b). This service is provided with a route finder service so that they can choose less-crowded trains if needed. In this way, we believe our method helps smarter railway trips in future ubiquitous society.

**CONCLUSION**

This paper proposed a method to estimate car-level positions of passengers on trains and car congestion levels to support smart urban mobility. Considering train specific mobility features, we have designed a novel algorithm for continuous update of likelihoods which represent car-level node positions and congestion levels in cars. The experimental results using real data have shown that the proposed method can estimate the positions of 16 passengers with the accuracy of 83% and the 2-level congestion with the F-measure of 0.82.

**REFERENCES**


6. Choset, H., and Nagatani, K. Topological simultaneous localization and mapping (SLAM) : Toward exact


30. onebusaway.org. The onebusaway project.


