Crowd Density Estimation in Railway Stations

Moustafa Elhamshary, Akira Uchiyama, Hirozumi Yamaguchi, Teruo Higashino[1,a)]

Abstract:
We present *CrowdApp*: a participatory system that leverages the sensed data collected from users' phones during their daily train commutes to gauge the real-time congestion level in railway stations. *CrowdApp* tracks the passenger's position in the station as well as identifies his/her context (e.g., waiting for a train, etc.). Therefrom, *CrowdApp* extracts novel features, based on the user's location and context, from the phone sensors to identify the surrounding congestion level in railway stations. Evaluation of *CrowdApp* through a field experiment in eight train stations shows that it can infer the congestion levels efficiently, highlighting its promise as a ubiquitous travel-support service.

1. Introduction

In highly populated cities, trains are the best-traveling option (e.g., 79% of commuters in Tokyo use trains [1]) as others, like buses, are often suffering from heavy traffic jams. Consequently, the traffic volume on certain trains is so intense (e.g., train cars' capacity reaches to 250% on average during morning rush hours compared to their typical capacity [2]). Even worse, the problem is acute in developing countries to the extent that train doors never close in some rail lines leading to killing a commuter every day [3]. Accordingly, overcrowding creates high levels of discomfort making train riding becomes harder for many passengers (e.g., pregnant women, handicapped people and parents with infants). Thus, many travelers may decide their preferable routes based on comfort than travel time [4].

The spatiotemporal analysis of railway stations' congestion level shows two peaks in weekdays at morning (8.00AM) and afternoon (5.30PM) as people have to arrive/leave their work by a certain time [4]. However, during peak times, the congestion level variability are higher than other daytimes due to some unpredictable factors (e.g., train delays, sports events, etc)[4]. In addition, the congestion peaks at some stations (i.e., higher than normal) do not necessarily mean that they are congested. Furthermore, at afternoon, the congestion level is not as steep as the morning (people do not have obligations on the return time to home). On the other hand, at weekends, the congestion level has a very high standard deviation, suggesting that it will not be as easy to predict [4]. Finally, as quantified in our survey study, the congestion level is not consistent either across all platforms in the same station nor across all waiting lines (i.e., train cars) on the same platform. Thus, the knowledge of congestion level beforehand will enable a myriad of **potential applications** like advanced station navigation which optimize passenger's route during the trip by rerouting her alternatively when a congestion is detected (e.g., avoid crowded cars and/or platforms) in large hub stations. Secondly, trip planning where passengers are urged to travel from/to a different but geographically close less crowded stations (i.e., congestion-based public transit selection). Thirdly, prevention of crowd accidents by asking passengers to change their travel time (i.e., travel slightly earlier or later). Finally, the congestion information will allow for actual travel-time prediction and evacuation guidance strategy for disaster control.

To alleviate the congestion problem, some train operators have attempted to discourage peak-time travel by increasing fare of peak-time than normal one [5]. However, this has had no observable effect on the congestion level at peak times [6]. Emerged congestion estimation Apps require passengers to manually report their travel experience which is invasive and susceptible to attacks [6]. Ideally, a congestion level estimation solution should meet/address the following criteria/challenges: (1) Ease of deployment: leverages existing infrastructure without installing special sensors. (2) Scalability: works across a large number of stations with minimal configuration. (3) Energy efficiency: avoids drainage of the phone battery (4) Minimal intrusion: minimal involvement of the user while preserving anonymity and privacy. (5) Accuracy: a fairly good estimate is necessary. (6) Low participation: no prerequisite on the active users number.

To tackle this problem, bountiful approaches have been proposed. However, they don't satisfy one or more of the previous criteria. Some techniques require specialized hardware for crowd monitoring (e.g., CCTV) [7], [8] limiting large scale deployment and may breach passengers' privacy. Existing phone-based techniques, e.g., [9], [10], prerequisite a considerable penetration ratio which is not predictable at the implementation time. This sparks the need for stations congestion level estimation system that meets/addresses the previous requirements/challenges.

In pursuit of this goal, we propose *CrowdApp* as a participatory system that gauges the congestion level in railway stations with minimal active users (as few as one). *CrowdApp* tracks the user's location along her trip in the station from the entrance. Simultaneously, it extracts different features from phone sensors to detect the passenger's state (e.g., walking, etc.). The passenger's detected states are forwarded to an inference algorithm employing Hierarchical Hidden Markov Model (HHMM) [11] to infer higher level passenger's activities (e.g., buying a ticket, etc.) while mitigating the human behavior artifacts effect and thus making the system robust to different

[1] Grad. Sch. of Info. Sci. & Tech., Osaka University
[a)] m.elhamshary, uchiyama, h-yamagu, higashino@ist.osaka-u.ac.jp

©2016 Information Processing Society of Japan

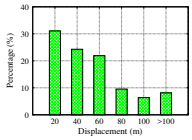


Fig. 1: Passengers average walking distance on platforms.

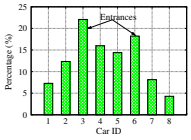


Fig. 2: Passengers average distribution over train cars.

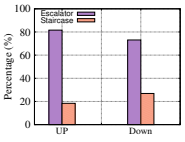


Fig. 3: Passengers preference of platform accesses.

users and stations. To address the challenge of low users participation, we extract novel features from sensors data based on the user's location (platform versus passageway) and context (walking versus queuing) to characterize the congestion level. In particular, the harnessed features draw on different passenger's behaviors and different characteristics of the ambient environment at different surrounding congestion levels. Finally, we implemented *CrowdApp* on Android phones where sensors are sampled in energy efficient manner, the data processing are guaranteed to preserve the user's privacy, and the evaluation results show that it can estimate the congestion level efficiently. In summary, our contributions are three-fold:

- We conduct a study of passengers' behavior in railway stations that reveals interesting findings regarding passengers' crowdedness pattern in railway stations (Section 3).
- We present the *CrowdApp* system that extracts novel feature from phone sensors, based on its context and location, to automatically and unobtrusively gauge the congestion level without having a prerequisite either on the number of participatory users or infrastructure support (Section 7).
- We collect real-time data by 16 participants, implement *CrowdApp* on Android phones, and evaluate it in 8 stations in 2 different cities (Section 8).

2. Related Work

Crowd density estimation is extensively studied research area. Many CCTVs have already been installed in railway stations for security. However, vision-based crowd estimation techniques are not effective due to occlusion, lighting condition and large-scale deployment cost. In addition, CCTVs are operators' property and they do not either share their videos (privacy issue) nor report the estimated congestion, if any, in public [12]. As smartphones are becoming increasingly powerful, their equipped sensors become a more cost effective approach to crowd density estimation. Refs. [9], [10] leverage audio tones to count the number of people in a place as it closely estimates the size of the crowd. To achieve this, [9] infers the number of active speakers in a dialog from audio data recorded on users' phones. However, it requires all passengers to speak while the silence is predominant in stations. Ref. [10] mines the number of mobile phones in a crowd by exchanging audio beacons among neighboring phones. However, it can recognize only phones participating in the crowd counting service and stations are filled with noise (especially at platforms) that will overlap with the audio beacon. Moreover, Ref. [13] show the possibilities of flock/group detection. However, it requires all group members to be system users and they have to swarm together into the station which is not practical. Ref. [14] sniffs WiFi packet transmitted from the surrounding phones to count the number of unique MAC addresses overheard and thus the number of people in the area. However, this scheme breaches the user's privacy (packet sniffing is almost illegal everywhere) as well as the need of phone kernel adaptation. Ref.[15] counts the number of devices leveraging Bluetooth collaboration by measuring RSSI fluctuation. However, the accuracy significantly depends on the number of Bluetooth-enabled phones. Finally, Ref. [16] estimates train car-level congestion leveraging Bluetooth signal strength interchanged among boarding passengers. However, it is infeasible for our problem as quantified in the evaluation section. In summary, existing phone-based techniques do not satisfy one or more of design criteria and substantially have different approach and goal. *Our target, however, is to estimate the congestion level without reliance either on users count nor a special infrastructure.*

3. The passengers' behavior survey study

To inspire the design of our work, we conduct a survey study on passengers' behavior at railway stations. We videotaped 19000 passengers trips from stations' entrance to platforms accommodating local trains (with non-reserved seats so their congestion level is unpredicted) at 12 different days and times and inspected them manually for analysis. Our study intends to concentrate on 3 questions: 1) which station's areas are usually congested?, 2) what is the distribution of waiting passengers along the platform and whether it is related to the location of the platform accesses (escalators, staircases, elevators)?, and 3) how are passengers distributed among platform accesses?

Regarding the **first** question, station's areas where passengers compete to acquire different facilities (e.g., elevators, platforms, ticket vending machines, staircases, escalators, trains, restrooms) are usually congested than the others.

To answer the **second** question, we analyzed passengers' trajectories on the platform. Figure 1 shows that the majority of passengers (76%) walk a short distance (less than 60m) on the platform to join a waiting line. The passenger's selection of a line depends on many factors including the elapsed time between the passenger and train arrivals to the platform, passengers' information and experience, the platform layout, among others. We noticed that waiting lines occupancy passes through three phases. Passengers reaching the platform early, finding most lines approximately empty, will mostly join lines optimized for short walking distance. Secondly, passengers reaching later, finding the nearest lines are a bit crowded, will tend to join the next available less crowded lines. Finally, passengers reaching shortly before the train arrival will join the nearest lines (even if crowded) to catch the train. Nonetheless, some experienced passengers may board cars near to their exits at the destination station [17]. In addition, the number and physical location of platform accesses have an impact on the passengers' distribution. Figure 2 shows the average passengers' distribution per train car on a platform with 2 entrances where passengers follow a skewed distribution (more dense near platform accesses). This is due to the short headway time of local trains (3 minutes on average) where passengers do not walk further to a convenient line as they do not wait a long time on the platform. However, in a multiple-accesses platform, the passengers' load will be dispersed among different lines (i.e., cars). Finally, the nonuniform passengers' distribution across station's street entrances, where passengers flow densely from main entrances (e.g., near shopping area or bus stops) than the others [18], also contributes to the skewed passengers' distribution near the ac-

©2016 Information Processing Society of Japan

cesses connected to main entrances.

For the **third** question, we found that most passengers prefer escalators than the collocated stairs (Fig. 3) due to passengers concern about the climbing effort. The number of passengers opted to move up using escalators (81%) is more than when moving down (73%) due to the fact that the effort in walking down a stairway is perceived to be less than that for climbing it up. In both cases, we noticed that passengers opted to climb stairs mostly when there is a potential delay in using crowded escalators. Elevators move a small number of passengers only.

Summary: Our study highlights that the congestion level is not uniform across the station and its estimation has the potential to enhance the passengers' experience and reduce their wait time (in crowded stations, they may wait for several trains before boarding [19]), enabling the next generation of travel-support system.

4. System Overview

As the user reaches the station, *CrowdApp* collects sensors information (gyroscope, accelerometer, magnetometer, barometer and microphone). The raw sensor measurements are preprocessed to reduce the effect of (a) phone orientation changes by transforming the readings from the mobile coordinate system to the world coordinate system [20] and (b) noise by applying a low-pass filter to raw sensor data. Given the privacy implications of turning on the microphone, we grant users full control over their sensed audio data where, despite natural concern, most users (71%) allow sound sensing in crowdsensing[21]. Additionally, we process **audio data locally** on users' devices to completely preserve their privacy and thus incentivize them.

We employ the TransitLabel system[22] which adopts Dead-Reckoning (DR) to tracks the passenger's position in the station due to its high accuracy, low-energy consumption, and reliance only on the phone sensors. It is designed specifically for stations where it leverages **unique** physical points (ticketing machines, fare gates, etc.) to alleviate the accumulation of the displacement error of DR. TransitLabel provides not only the user's physical location (X ,Y) but also her logical location (platform, passageway, etc.) as it builds stations indoor floorplan highlighting different landmarks (ticketing machines, fare gates, etc.).

The **Passenger's State Detection Module** will detect the passenger's state (e.g., walking, etc.) along her trip in the station. The sequence of the detected states is fed to the **Passenger's High-level Activities Inference** which incorporates the HHMM to infer the most likely sequence of passenger's high-level activities (e.g., buying a ticket, etc.). The recognized activities are harnessed to refine the user's location as each activity (e.g., buying a ticket) is uniquely associated with a landmark (ticketing machine).

Finally, the passenger's context and location are forwarded to the **Congestion Level Estimation Module** which, based on them, extracts a novel set of features to characterize the surrounding congestion level at the most susceptible areas to passengers' crowdedness. In particular, in crowded stations, passengers line up for different facilities (e.g., ticketing machines, elevators, etc.) where their waiting time and order in the queue are used as the temporal/spatial features to infer the congestion level on the nearby of these facilities. In addition, the passenger's walking behavior (e.g., walking speed) and the ambient sound at walking areas (e.g., passageway) could reflect the surrounding congestion level. An identifiable signature of rapid change rate and heavy distortion surfacing on the barometer and magnetometer values respectively can characterize the congestion at escalators and elevators. Finally, the user's spontaneous locations and activities could be employed to estimate passengers' flow on the platform.

Sensor	**Feature**	
Accelerometer	Time domain	mean, STD, variance, magnitude, correlation, min, max, range, steps
	Frequency domain	FFT-Peak, spectral Energy and Entropy, FFT DC,1,2,3,4,5,6 Hz
Gyroscope	Time domain	mean, variance, gradient, min, max
Magnetometer	Time domain	variance, max, min, range, gradient, peak length and intensity
Barometer	Time domain	variance, gradient, mean, min, max
Audio	Time domain	normalized occurrence count
	Frequency domain	512pt FFT, FFT-Peaks

Table 1: The extracted features list for each sensor.

5. The Passenger's State Detection

To detect the passenger's state, we extract different features from sensor data as shown in Table 1 where the frequency domain features are not extracted for some sensors as they proved ineffective. To be robust, we use offset-independent (e.g., variance) and orientation-independent (e.g., magnitude of acceleration and magnetic field) features. The set of the extracted features (Tab. 1), except audio data, and the corresponding passenger's states from a large dataset collected in our preliminary experiments are used to train a Random Forest based classifier [23] offline. Later at run time, the classifier can infer the passenger's state from the real-time extracted features. The detected passengers' states include: *A. Motion States*

We extract different features (Tab. 1) from the accelerometer to identify the following states (standing, sitting, stepping, normal walking, slow walking, climbing, accelerating, decelerating).

B. Change Altitude

To identify it, we adopt the technique in [24] that is based on the difference among the relative barometer readings (i.e., pressure) in consecutive overlapping windows. This is essential to separate the usage of elevation change elements (elevator, escalator or staircase) from other activities (e.g., buying a ticket, etc.).

C. Sense a Magnetic Distortion

Nowadays, many electronic machines are installed in railway stations (e.g., fare collection gates, etc.). The direct passenger interaction with these machines distorts the sensed magnetic field reading by machines' metals and electronics. This distortion forms a peak in the magnetic field readings where the peak duration and strength are extracted to identify this state.

D. Change Heading

Station installed machines (e.g., vending machines) are usually mounted to walls, so their usage (e.g., buying tickets) involves a sudden change in the user direction (i.e., surge in gyroscope readings) directly after the activity. The derivative of gyroscope values within a time window is used to detect this sudden change.

E. Hear a Unique Sound

We adopt the technique in [22] to detect the drink and ticket vending machines based on the unique sounds (drink falling and beep sounds respectively) they emit during the user interaction.

©2016 Information Processing Society of Japan

Observation (user state)	Possible Hidden state (Context)
Change altitude	Move from a floor to floor
Change heading	walk away after using a machine, join a waiting line, etc.
Near a magnetic distortion	Interacting with an electronic machine, being on an escalator, etc.
Hear a unique sound	Interacting with a vending machine
Stand	using a machine, Waiting for a train or in a queue, etc.
Walk	Walking at a walking area, walking into a train, etc.

Table 2: Examples of passenger's states and their contexts.

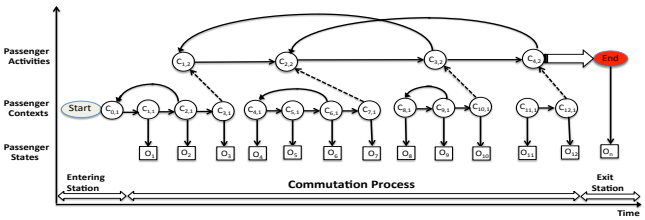


Fig. 4: The model of the whole commute process.

F. Be in a Train

We follow a hybrid approach fusing different sensors (accelerometer [25], barometer [26] and magnetometer [27]) to provide a high accuracy detection of being on a moving train. In addition, the algorithm in [27] is adopted to detect train stops.

6. The Passenger's High-Level Activities Inference

This module receives the sequence of the user's detected states by the previous module and leverages it to recognize her activities. A passenger's activity is composed of a temporal sequence of micro-activities (i.e., states). For example, buying a ticket consists of the following sequence of states. First, the user has to be stationary while buying a ticket from the vending machine. Second, the phone senses a magnetic distortion and overhears a distinct beep sound during the user interaction. Finally, the user has to change his direction after finishing machine interaction to resume walking. In addition, the overall commute process is a sequence of temporally correlated in-station activities. An example of a whole commute process consists of the following activities sequence: buy a ticket, cross a fare gate, walk at a passageway, access the platform by an escalator, walk at the platform, wait in line for the train, and board the train. Thus, we employ the HHMM (Fig. 4) to infer the most likely sequence of passenger's activities constituting her train commute from the detected states sequence. HHMM allows to capture the idea of the whole commute process instead of a sequence of unrelated activities. The HHMM is invoked when the context (Start-Commute) is signaled as the user enters the station. The inference process ends signaling (Finish-Commute) after the passenger gets off the train.

We use a two-layer HHMM where in the bottom layer, the relation between the observation (passenger's state) detected from the sensor data and the passenger's context (hidden state) is modeled as an HMM with standard first-order Markov assumptions. As HMM factors in the temporal correlations between consecutive passenger's contexts, it may potentially counteract the errors in classifying the passenger's states by inferring the most likely sequence of contexts that corresponds to the detected state sequence. For example, if the phone is in the state of sensing a magnetic distortion, it may not definitely mean that the user is interacting with a vending machine. Instead, the user may be on an escalator, on the nearby of other electronic devices, or it may be a noise. So, depending on the given sequence of states, HMM will infer the most likely corresponding context (Tab. 2). The emission probability for a given *<state, context>* pair represents the probability of seeing that context conditioned on the user being in that state. The transition probability is the probability of transitioning from one context to another. The upper layer of HHMM, modeled as a Markov chain, will infer the most likely sequence of passenger's activities where each activity is identified from a temporal sequence of user's context detected at the lower HHMM layer (Tab. 3).

Activity	States
Use an elevator	normal walk, stand, slow walk, direction change, floor change, accelerate, stationary, decelerate
Stand on escalator	stand, floor change, sense distortion
Climb a half-landing stair	climb, floor change, direction change, walk, floor change, climb
Climb an escalator	climb, floor change, sense distortion
Climb a straight stairs	climb, floor change, no magnetic distortion, no direction change
Use a drink vending machine	no floor change, stand, sense distortion, drink falling sound, direction change
Use a ticket vending machine	no floor change, stand, sense distortion, beep sound, direction change
Cross a gate by a ticket	no floor change, normal walk, decelerate, slow walk, sense magnetic distortion, accelerate, normal walk
Cross a gate by an IC card	no floor change, normal walk, decelerate, slow walk, pause, sense distortion, accelerate, normal walking
Queue	no floor change, stand, step forward, stand, step foward,....
Wait for and board a train	normal walk, stand, head towards platform, step aside, stand, walk, stand, be on a train
Get off a train	be on a moving train, train stops, normal walk

Table 3: The passenger's states used in activities recognition.

We learn the HHMM parameters offline based on a collected training dataset by applying the Expectation-Maximization (EM) algorithm [28]. As EM converges to local maxima and searching for global maxima in the likelihood landscape is intractable, we used the Gibbs Sampling [29] technique for a more robust estimation. To derive the initial values for the stationary, transition and observation probabilities, we apply statistics on our dataset used for training. To apply this model for identifying the most likely sequence of high-level passenger's activities from a sequence of observation, we use the Viterbi algorithm [30]. Now, the list of recognized passengers' activities includes:

A. Using an Elevation Change Element

In this class of activities, users move from one floor to another. We adopt the idea in [31] to classify their fine-grained activities (using elevators, escalators, and stairs) as shown in Tab. 3.

B. Using a Ticket or a Drink Vending Machine

Their typical usage trace consists of normal walking to the machine, standing in front of, inserting currency, beginning the service (choosing a drink or ticket type), finishing the service (grabbing the ticket or the drink), and finally walking away. To discriminate the two vending machines classes, we draw on which unique sound (drink falling, beep) is detected from the audio recorded during the machine interaction [22].

C. Crossing a Fare Collection Gate by Tickets or IC Cards

As the passenger approaches a gate, a noticeable slows down in her walking speed is observed till she pauses by the gate to drop the ticket into or tap the IC card over the machine, and then she resumes walking. Simultaneously, crossing the gate heavily distorts the mag-

©2016 Information Processing Society of Japan

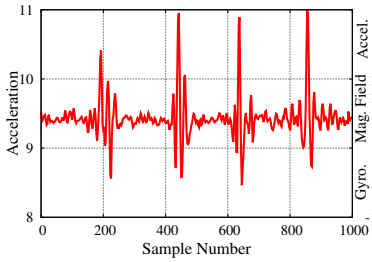


Fig. 5: The queuing pattern.

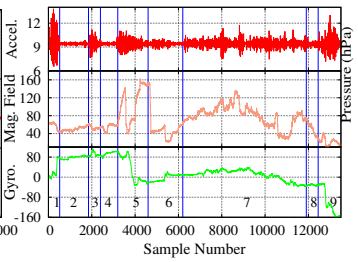
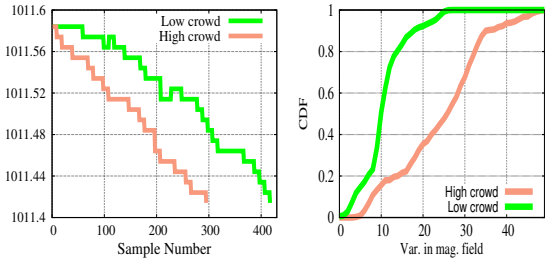


Fig. 6: The train commute pattern.



(a) Barometer. (b) Mag. field variance.

Fig. 7: Sensor readings at two escalator congestion levels.

netic field by the gate's ferromagnetic metals.

D. Using a Restroom

We incorporate the algorithm in [32] which actively probe the acoustics of environment with the phone's built-in speaker and microphone and analyze the impulse response (IR) to detect the type of space (restroom or not).

E. Queuing in Front of a Facility

In a crowded station, passengers normally line up for different facilities. The typical queuing behavior for elevators, ticketing machines, and restrooms (i.e., with long service times) takes the form of a repetitive sequence of standing interspersed with short bursts of stepping forward states (e.g., 4-episodes pattern is shown in Fig. 5) till the user dequeues. Once the user dequeues and allocates the facility, its unique sensor pattern will surface on the phone sensor, so we associate each queuing activity with a certain facility. On contrast, users will continuously walk forward slowly with short time pauses when they line up for escalators as they carry more people. To accommodate the variability in queuing patterns among different facilities, we run concurrent motion classifiers with different window sizes to detect queuing.

F. Being on Platform and Boarding a Train

The platform is the area where passengers board trains during their train commute. As shown in Fig. 6, the user's motion state on the platform changes during a short time period from (1) walking on the platform, (2) joining a line, (3) (4) short walking with slight heading change and then standing (step aside making the departure route available for alighting passengers), (5) walking into the train car, (6) waiting the train to move, and finally (7) being on a moving train. This short temporal sequence of states is leveraged to detect the platform track position (estimated from the passengers' positions during the train boarding).

Once the platform area is identified, when the passenger is being on the platform and its state is switching from walking to standing while turning to face the platform track (Fig. 11), this means that the user joined a waiting line. Later, lines position on the platform can be estimated from the passengers' positions reported while they are waiting for the trains on the platform by the DBSCAN clustering approach. Once lines are detected, every neighbored collection of n lines (n is the number of doors per car, which is a known constant) is representing a queuing area for a train car.

G. Getting off a Train

It is detected from a sequence of context changes from being on a moving train to a stopped train to walking (Fig. 6, Sects. 8,9).

7. The Congestion Level Estimation

This module extracts different features from sensors data based on the user's location and context (e.g., walking on the platform) retrieved from the previous modules to characterize the surrounding congestion level. The congestion indicator function differs from occasion (location and context) to another where, for example, it may be the queuing time for facilities and the passengers flow on platforms. However, at all occasions, we map the congestion indicator function to a coarse-grained congestion level (low, medium, high) which is reported to the users. We estimate the congestion level across the following locations:

A. In Front of Facilities

In the steady state, the location of stations' facilities (ticketing machines, restrooms, elevators, etc.) will be already identified from users' locations during their usage[22]. *CrowdApp* tracks the user's location and as soon as she becomes stationary in the vicinity of one of the facilities, the enqueuing event is triggered. Whenever the pattern of the designated facility surfaces on the user's trace identified by the instants of level changing (for elevators and escalators), magnetic distortion (for ticketing machines) and the active probing signal (for restrooms), it marks the start of the service time and the dequeuing time. When the user walks away from the facility, it marks the end of the service time. Given that the context changing (enqueue, dequeue, allocate and release the facility) times are recorded, we can directly derive the queuing (waiting) and service times. Although the service time does not contribute to the congestion level estimation, it could be leveraged in the average travel time computation. The number of consecutive stepping forwards episodes during queuing can allude to the passenger's order at her enqueuing time. Eventually, the waiting-time and the order in queue represent the temporal/spatial features that are mapped to a congestion level category (low, medium, high) arbitrarily by the user and vary from facility to another (e.g., we use 2, 4 as queue length thresholds for low, medium and high levels for ticketing machines).

B. On Escalators/ Elevators

As most passengers access the platform by escalators, they are the most crowded elevation change element. Additionally, as passengers tend to join the nearest less-crowded lines, so gauging the congestion level on escalators is indicative of the congestion on the connected platforms' part. Many manufacturers (e.g., [33], [34]) have developed new variable speed escalator/elevators models (Eco machines) that adjust their speed based on passengers' load. The escalator automatically slows or halts when it carries fewer passengers or no pas-

©2016 Information Processing Society of Japan

sengers respectively to save energy. The escalator operates at a low speed in stand-by and gradually increases speed to the rated speed after detecting a passenger approaching the boarding area by using weight sensors. This model is ideal for railway platforms where passengers' flow tends to be intermittent, so they are in the market since 2008 [35]. Figure 7a shows that the time an escalator takes to move from one height to another is contingent on the congestion level. To recognize whether the escalator is congested (carries more than 65% of its maximum capacity) or not, we draw on its speed of movement measured by the rate of pressure change (i.e., barometer readings) during its usage time as illustrated in Algorithm 1.

In addition, for escalators to maintain a fast speed while moving

Algorithm 1 Estimate Escalator Congestion

Input: $\mathbf{p}$ ▷ Pressure values during escalator use
Input: $\mathbf{m}$ ▷ Magnetic magnitudes during escalator use
Output: CL ▷ Escalator congestion level
  $\Delta p \leftarrow$ PressureChangeRate($\mathbf{p}$)
  if $\Delta p >$ TH$_p$ **then**
    $CL_p \leftarrow$ high
  else
    $CL_p \leftarrow$ low
  end if
  $V \leftarrow$ VarDistribution($\mathbf{m}$, 4sec) ▷ Variances estimated every 4 secs
  if $KL(V|K_{low}) > KL(V|K_{high})$ **then** ▷ Variances distributions similarity
    $CL_V \leftarrow$ high
  else
    $CL_V \leftarrow$ low
  end if
  CL=Fuse (CL_p, CL_V) ▷ Fuse both estimations

Algorithm 2 Fuse Congestion Estimation Results

Input: CL_A, CL_B ▷ Individual congestion estimation results
Output: CL ▷ Final estimated congestion
  if $CL_A = CL_B$ **then**
    $CL = CL_A$ ▷ Agreed levels
  else if CL_A = medium OR CL_B = medium **then**
    CL = medium ▷ One is medium an the other is low or high
  else ▷ contradicted levels
    CL = unknown
  end if

a large number of passengers, it has to exert more force (i.e., energy). This causes a large distortion of the ambient magnetic field (by escalator machinery) which can be leveraged to estimate the load on the escalator. Figure 7b shows the cumulative distributions of the magnetic variance values for data collected in our preliminary experiment. In low congestion level cases, more than 80% of the variance samples are concentrated below 15. This is in contrast to the high congestion level where the variance samples are distributed over a much wider range. This inspires us to identify the congestion level by using the K-L divergence [36] technique to examine the difference in the distribution of the magnetic field changes. The KL divergence is calculated among the histograms of magnetic field variances collected while the passenger is on the escalator (V) against the magnetic fingerprints labeled with less or highly congested escalators (G_{low} *and* G_{high}) as:

$$KL(V|G) = \sum_i V(i) log \frac{V(i)}{G(i)}$$

The estimated congestion level (low, high) will be deemed as the one having the lowest K-L value (i.e., similar magnetic variance distribution) with the test magnetic field variance histogram as shown in Algorithm 1. We will fuse the barometer and magnetometer estimations to enhance the final congestion level estimation as shown in Algorithm 2.

Algorithm 3 Estimate Congestion around Path from p_s to p_e

Input: $\mathbf{L}$ ▷ Inertial sensor values from p_s to p_e
Input: $\mathbf{mic}$ ▷ Microphone amplitudes
Output: CL ▷ Area congestion level
  $S \leftarrow$ StepIntervalDistribution($\mathbf{L}$)
  if more than 80% of intervals in S < 0.6 sec **then**
    $\mathbf{t} \leftarrow$ UserTrace($\mathbf{L}$) ▷ Normal or fast walk
    if Distance($\mathbf{t}$, ShortestPath(p_s, p_e)) < TH$_d$ **then**
      $CL_L \leftarrow$ low ▷ Walk along with shortest path
    else
      $CL_L \leftarrow$ high ▷ Winding path
    end if
  else
    $CL_L \leftarrow$ high ▷ Slow walk
  end if
  $A \leftarrow$ NormalizedDist($\mathbf{mic}$) ▷ Normalized Occurence Count
  if ClosestDistribution(A) = A_{low} **then**
    $CL_{mic} \leftarrow$ low
  else if ClosestDistribution(A) = A_{med} **then**
    $CL_{mic} \leftarrow$ medium
  else
    $CL_{mic} \leftarrow$ high
  end if
  CL=Fuse (CL_L, CL_{mic}) ▷ Fusion

C. At Walking Areas

They include the mezzanine, passageways, and platforms. We harness the following two features:

Inertial Sensors-based Features Analysis of pedestrian movement shows a strong correlation between the walking speed and the surrounding crowd density. The average step interval (i.e., walking speed) tends to be longer with higher crowd density as pedestrians have to walk at a similar speed with the surrounding people (Fig. 8). We mine the interval among steps detected by [37] to identify the surrounding congestion level. If more than 80% of the step intervals in the window are less than 0.6 seconds, we regard that the user is walking at the normal speed and moving at a slow speed otherwise. However, some passengers walk at normal speed even in highly crowded areas resulting into some misclassifications. To mitigate this effect, we observed that passengers, in this case, have to change their heading frequently to maintain a normal walking speed while avoiding collision with nearby people (Fig. 9). Our hypothesis is that when the walking area is empty, the user trajectory will theoretically be a straight line. Thus, we break the user's trace into multiple trajectories where each one comprises a part of the trace from one landmark to the next in the route (e.g., from the ticketing machine to the gate). Then, we calculate the *trajectory stretch factor* as the Euclidean distance between the corresponding points in the shortest and user actual trajectories. Finally, the user's walking speed and her trajectory stretch factor are mapped to the congestion level empirically as shown in Algorithm 3. A walking region is low congested when the crowd density is lower than 1.0 person/m^2, medium when it is between 1.0 and 2.5 persons/m^2 and high otherwise (Fig. 12).

Audio-based Features

The basic idea is that the people crowd generates noise that could be identified from the ambient sound. To identify the congestion level from the ambiance, we employed a light-weight fingerprinting scheme (to avoid increasing the workload on the user's phone) to characterize each congestion level with a sound fingerprint. Fingerprints are collected regardless of the station or the walking area

©2016 Information Processing Society of Japan

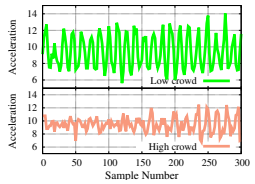


Fig. 8: Step interval at different congestion levels.

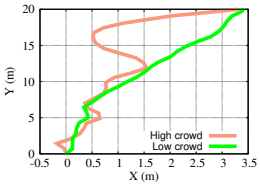


Fig. 9: User trajectory at different congestion levels.

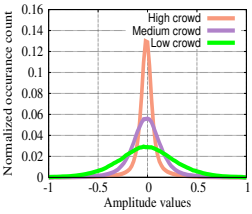


Fig. 10: Sound fingerprint versus congestion level.

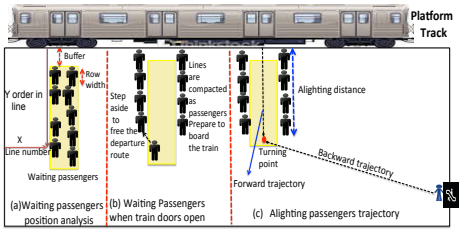


Fig. 11: Passengers behavior on the platform.

they collected at. Before fingerprinting, we extracted the frequency components below 2KHz from the audio recording which are found empirically to be effective in the congestion level estimation. Our fingerprinting scheme is based on the signal amplitude to capture the loudness of the sound [38]. Specifically, the amplitude is divided into 100 equal intervals and the number of samples per interval is normalized by the total number of samples in the recording. The 100 normalized values are considered as the feature characterizing the congestion level. Figure 10 shows the fingerprint for three different congestion levels. At run time, we compute the Euclidean distance among the sound fingerprints of the three congestion levels and the test signal as shown in Algorithm 3. The congestion level fingerprint showing the lowest Euclidean distance with the test signal is deemed as the congestion level. The audio-based and inertial-based estimated congestion levels are fused by Algorithm 2 to report the final congestion level.

D. On The Platform

The passenger's waiting location on the platform could identify the car she boarded as the X and Y coordinates of her position correspond to the waiting line (i.e., car and door numbers) and her location away from the platform track respectively (Fig. 11-a). So, the user order in the line (O) can be estimated as: $O = (Y - B_{size})/(R_{space})$ where B_{size} and R_{space} are the buffer size (the free space between the track and first row) and the row space (space allocated to each row) respectively (Fig. 11-a) and their values are fixed for all stations run by the same company. The user order in line alludes to the instantaneous line length at her enqueuing time. Since the user's line occupancy changes rapidly while waiting for train, the instant line length is infeasible to indicate the congestion. However, the arrival rate to the line can be estimated, if we have many users in the line, and thus indicating the congestion. As this violates our assumption of few users, we finally revert to estimate the passenger's flow to or from train cars to indicate the congestion level. To achieve that, we track passengers' activities on the platform. Specifically, when the train reaches, the waiting passengers step aside to free the departure route for alighting passengers as soon as it stops (Fig. 11-b). When the doors open, the alighting passengers walk across the waiting lines till they reach their end where they will make a turn (i.e., *the turning point*) to walk away (Fig. 11-c). After the last passenger departed the train, the waiting passengers will walk into the train. The time elapsed since the instant a waiting passenger waits after stepping aside to the time she started walking into the train, **alighting time**, is directly proportional to the number of alighting passengers. On the other hand, the straight distance walked (orthogonal to train movement direction) by alighting passengers on the platform before turning, **alighting distance**, is proportional to the length of line on the platform. However, sometimes we cannot separate the distance walked on the platform from that walked inside train as some trains doors are in the same level of the platform. To mitigate this effect, we track the alighting passenger till she reaches any of the platform accesses (e.g., escalator) which we maintain their locations. Then, we build a *backward trajectory* from the access point to the turning point location (Fig. 11-c). The alighting distance is the distance between the platform track to the turning point. Finally, we build two linear regression models to map the alighting time and distance to the corresponding alighting and waiting passengers' counts respectively which are in turn mapped to a congestion category (low, medium, high) by an empirical setting (e.g., high level is indicated if there are 14 or more waiting passengers).

8. Performance Evaluation

CrowdApp is evaluated through deployment at 8 stations by a group of 16 users collecting two datasets (scenario-based and free). In scenario-based, 10 participants were assigned specific trips from the station entrance to different platforms. The free dataset is collected by 6 individuals from their everyday train commute. We deployed two Android applications: a background data collection tool to sample all sensors and a foreground tool to manually record ground truth. The data collection was conducted at different times and days and using different Android phones including Samsung Galaxy S5 and LG Nexus 5. In addition, stations are in two different cities, managed by different companies and having different buildings designs and sizes. This captures the time-variant nature of the congestion level, the heterogeneity of users and devices and emphasizes *CrowdApp* scalability. The average platform dimensions are $180m \times 12m$. Train cars are of 20m length with 3 doors with average inter-distance of 5.2m.

A. Activity Detection Accuracy

Figure 13 shows that most user's activities can be identified accurately due to their unique sensor pattern regardless of the user/station. Detecting boarding and getting off trains are challenging due to the error inherent in the employed transportation-mode detection method. Queuing for long service-time facilities (elevators, restrooms, ticketing machines) can be detected in 100% of cases but it is more challenging for escalators as the congestion has to exceed a certain threshold to impede user's smooth mobility to be detected. In all cases, queuing detection incurs false positive samples. However, these errors can be mitigated by leveraging the user's and landmarks' location information as queuing be at certain locations, so we can filter out outlier activities happening far from the

©2016 Information Processing Society of Japan

(a) Low (b) Medium (c) High

Fig. 12: Snapshot of different congestion levels on the platform.

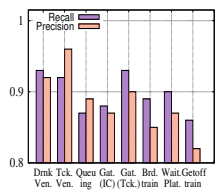


Fig. 13: Activity recognition Accuracy.

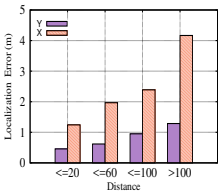


Fig. 14: The average user location accuracy on the platform.

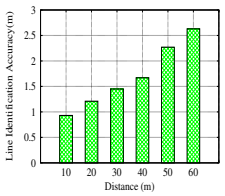


Fig. 15: The average waiting lines location accuracy.

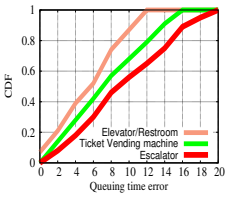


Fig. 16: The CDF of queuing time estimation error.

designated facilities.

B. Location Estimation Accuracy

Figure 14 shows that the user's position while waiting in line can be accurately estimated especially when users' lines are near to platform accesses (Fig. 1). This accuracy stems from the fact that platforms have many accesses (elevators, escalators, etc.) and whenever their pattern surface on the user's trace, the dead-reckoned derailed position will be calibrated to their access locations on the platform. Thus, the location error grows only when the user moves away from this access point on the platform.

Figure 15 demonstrates that the average waiting lines location accuracy relies on their distance to the nearest platform accesses (location is more precise for lines near to the platform accesses as most user trails are short). Generally, given that we can estimate passengers' location when they are waiting in lines accurately (Fig. 14) and the inter-line distance of 5.2m, so we can identify the waiting lines location and thus boarded cars accurately.

C. Congestion Level Estimation Accuracy

In front of facilities Figure 16 shows the accuracy in the waiting times error estimation which is up to 12, 16 and 20 seconds for elevators, ticketing machines and escalators where the extreme values are rare (the median error is 5, 7 and 9 seconds respectively). The enqueuing recognition is more challenging than the dequeuing especially for escalators where passengers do not have the normal queuing pattern. In this case, the enqueuing time is detected through at least 3 short windows of activity for verification. The dequeuing time, also, is a bit tricky for all facilities as there is a time elapsed to detect the marking signal of starting the service time (level change for escalators/elevators, magnetic distortion for ticketing machines, IR for restrooms). To mitigate this effect, we roll back to the trace and subtract this elapsed time (from dequeuing to the start of service time detection) from the estimated queuing time. The final congestion level (mapped from the waiting time) estimation accuracy is shown in the first (yellow) row in Table 6.

On escalators The confusion matrix (Tab. 4) shows that the barometer-based congestion estimation can detect the high and low level correctly in 76% and 79% of cases. However, this approach cannot be applied when the user climbs the escalator as the rate of change of barometer values will be affected the climbing speed of the users. Nevertheless, the magnetometer-based method can be applied in all cases disregarding the passenger motion state (standing or climbing) as the magnetic distortion, by the escalator machinery, is persistent. However, it sometimes estimates the congestion level incorrectly due to some overlap in the magnetic variance between low and high crowd densities (Fig.7b). The fusion of the two reported congestion categories provides a better congestion level estimation of 82% on average.

At walking areas The confusion matrix (Tab. 5) shows that acceleration-based method can correctly identify the high congestion level in 79% of cases. We observed that users sometimes walk at normal speed without snaking in narrow crowded areas when nearby people flow in the same direction (no intersections with opposing passengers). This leads to some high congestion misclassification and restricts the system ability to separate medium and low congestion level classes. Nevertheless, the low/medium, as one class, congestion level can be estimated correctly in 83% of cases as passengers do not slow down their walking unreasonably. The errors in low/medium density estimation mainly happen at upslope areas where users may naturally slow down the walking speed regardless of the surrounding congestion levels.

The confusion matrix (Tab. 5) shows that audio-based congestion level estimation achieves an average accuracy of 77%. In the case of low congestion, *CrowdApp* achieves lesser accuracy (especially in platforms) due to the surrounding noise that is not caused by a human crowd (e.g., announcements, human speech, trains on opposite platform). While we trained the audio-based classifier by aggregating all the audio samples regardless of the area where they are recorded, the accuracy of the classifier would be further improved by constructing tailored classification models for each area. However, it incurs heavy processing and power consumption that does not fit the phone limited resources.

Finally, fusing the two methods could gauge the congestion level with an accuracy of 76-87%.

On the Platform Figure 17 reports the accuracy in detecting the user order in the line. In 76% of users' traces (i.e., users walk <60m on the platform), we can identify the user order in line with +1/-1 error in 63% of cases. Although the user's location while waiting in line can be estimated accurately (Fig. 14), the order in line have errors which are mainly due to the nonuniform space allocated by each row of passengers (Fig. 11).

Figure 18 quantifies the CDF of the estimation error of the number of boarding and alighting passengers. It shows that the error in estimating the number of waiting passengers (up to 4 in 82% of cases) is lesser than the estimation of alighting passengers (up to 4 in only 61% of cases). The estimation error of the alighting distance (mapped to the number of waiting passengers) is due to users' step size differences, steps miscount and some users do not turn directly after reaching the last passengers in the waiting line. However, this error is empirically bounded by 2.4m which is mapped to around 3 rows (6 passengers) and does not happen frequently. On

©2016 Information Processing Society of Japan

	Barometer		Magnetic		Fusion		
	Low	High	Low	High	Low	High	Unknown
Low	**79%**	21%	**77%**	23%	**84%**	12%	4%
High	24%	**76%**	26%	**74%**	14%	**80%**	6%

Table 4: Accuracy of congestion level estimation on escalators

	Acceleration			Audio			Fusion			
	Low	Med.	High	Low	Med.	High	Low	Med.	High	Unknown
Low	**83%**		17%	**73%**	20%	7%	**87%**	7%	3%	3%
Med.				9%	**76%**	15%	6%	**76%**	10%	8%
High	21%		**79%**	4%	13%	**83%**	3%	6%	**86%**	5%

Table 5: Congestion level estimation accuracy at walking areas.

	Act. Recog.	Congestion Estimation (%)											
		Facilities			Escalator		Walking			Platform			
		L	M	H	L	H	L	M	H	L	M	H
Database	90	94	89	88	87	86	90	84	89	83	86	82
Validition	86	91	90	86	80	77	87	81	85	80	85	80

Table 6: *CrowdApp* performance (measured by F-Measure [39])

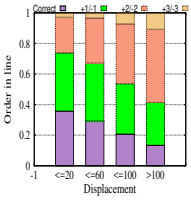


Fig. 17: The user order in line identification accuracy.

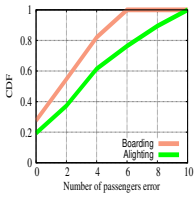


Fig. 18: The CDF of passengers flow estimation error.

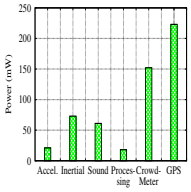


Fig. 19: The Energy footprint of *CrowdApp*.

contrast, the alighting time may be inaccurately estimated as some alighting passengers (e.g., parent with infants, handicapped people and people with luggage) spend more time to get off the train affecting the estimated alighting time. However, we believe that the number of alighting passengers (within 4 passengers errors in 61% of cases) is acceptable as our goal to infer the congestion level not to count boarding and alighting passengers. The final congestion level (mapped from alighting and waiting passengers count) estimation accuracy is shown in the first (yellow) row in Table 6.

D. Power Consumption

Figure 19 shows the power consumption of *CrowdApp*. To illustrate the contrast in power consumption, we run a 1-second GPS sampling APP (***GPS is not available in stations but it is used as a baseline***). The power is calculated using the PowerTutor [40] and the Android APIs using the HTC Nexus One cell phone. *CrowdApp* leverages inertial and sound sensors for its operation. The inertial sensors are always running to detect the phone orientation changes and for indoor positioning which are basis functions for a variety of mobile applications. So, *CrowdApp* typically consumes a little **extra** power of 79mW from the sound sensor and its local data processing. The power consumption of *CrowdApp* is significantly (32%) less than the GPS consumption which will not impede the user participation to the system.

8.0.1 Generalization of *CrowdApp*

E. Generalization of CrowdApp

We based the activity recognition and congestion level estimation on the majority of commuters typical behavior and typical environment characteristics. However, some users may have different behaviors and some stations have different building structures and machines hardware that may lead to some activity and/or congestion misclassifications. Thus, to demonstrate that *CrowdApp* could generalize over various users and stations, we show a one-station-out cross-validation (one station data from the free dataset is selected as the validation data while all data collected at other stations is used as the database) results in Table 6. The table compares the F-measure [39] of the activity recognition and congestion estimation of the database against the one-station-out data. The results demonstrate that *CrowdApp* performance does not deviate much between different stations/users (escalators have the worst deviation due to hardware differences) due to the uniform nature of passengers' activities at stations and fusing multiple features to identify activities and congestion level, reducing the sensitivity to specific user or machines.

9. Discussion

Impact of phone pose Experiments carried out in different phone poses (hand, pocket) suggest that the accuracy is not significantly dependent on the device pose. This robustness may due to the transformation of sensor readings to the real world coordinates, data smoothing and extraction of pose-independent features (e.g., magnetic field). Additionally, the maximum in-pocket discrepancy in audio spectrum values is only 0.07dB on average.

Congestion variance We rely on a few number of users (as few as one) which may report a spontaneous anomalous (spike or false) congestion level. The increase in the users' count will mitigate this effect. If there are many users reporting congestion of an area, we select the congestion level based on the majority vote. If an area had been reported as congested, the event can be removed if multiple following users report contradicted level or it will be set to unknown after a certain time frame which is closer.

User incentive Users are naturally incentivized to get the congestion information. To increase their engagement, we can support social interactions where users see locations of nearby users and do chat with them which give users the sense of large community. We may offer users virtual rewards (e.g. mayorship, badges) which reflect their contribution based on other members feedback.

Cars congestion level It can be estimated by tracking the number of passengers boarding or alighting cars at each station along the train route. This is part of our future work.

Other environments *CrowdApp* can be customized to other environments, e.g., airports, that have similar facilities (e.g., elevators, escalators, vending machines, lockers, security gates, and boarding-pass printing machine).

10. Conclusion

We presented *CrowdApp*: a system that leverages the phone-sensed data to gauge the congestion level in railway stations offering a promising way to ease the congestion and improve the comfort level of passengers. *CrowdApp* can estimate the congestion level without much reliance on the number of participants or specialized hardware which is more scalable and robust.

References

[1] : The 5th Tokyo Urban Area Person Trip Survey, `http://www.tokyo-pt.jp/`.

[2] Yamakazi, F. and et al.: Measurement of the Congestion Externality in Rail Commuting in the Tokyo Metropolitan Area, Technical report (2006).

[3] : Train incidents at India., `http://www.worldcrunch.com/culture-society/mumbai-039-s-crowded-trains-are-literally-killing-commuters-ever`

[4] Ceapa, I. and et al: Avoiding the crowds: understanding tube station congestion patterns from trip data, *SIGKDD*, ACM (2012).

©2016 Information Processing Society of Japan

[5] : Congestion Charge Zone of Tfl London., https://tfl.gov.uk/modes/driving/congestion-charge/congestion-charge-zone?intcmp=2055.

[6] Lathia, N. and et al.: How smart is your smartcard?: measuring travel behaviours, perceptions, and incentives, *Ubicomp*, ACM (2011).

[7] : The LightHaus Visual Customer Intelligence., http://www.lighthausvci.com/.

[8] Teixeira, T. and et al.: Tasking networked CCTV cameras and mobile phones to identify and localize multiple people, *UbiComp*, ACM (2010).

[9] Xu, C. and et al.: Crowd++: unsupervised speaker count with smartphones, *Ubicomp*, ACM (2013).

[10] Kannan, P. and et al.: Low cost crowd counting using audio tones, *SenSys*, ACM (2012).

[11] Fine, S. and et al.: The hierarchical hidden Markov model: Analysis and applications, *Machine learning* (1998).

[12] Lathia, N. and et al: Tube star: crowd-sourced experiences on public transport, *MOBIQUITOUS*, ICST (2014).

[13] Kjærgaard and et al.: Detecting pedestrian flocks by fusion of multi-modal sensors in mobile phones, *UbiComp*, ACM (2012).

[14] Chon, Y. and et al.: Sensing WiFi packets in the air: practicality and implications in urban mobility monitoring, *Ubicomp*, ACM (2014).

[15] Weppner, J. and et al.: Bluetooth based collaborative crowd density estimation with mobile phones, *PerCom*, IEEE (2013).

[16] Maekawa, Y. and et al.: Car-level congestion and position estimation for railway trips using mobile phones, *UbiComp*, ACM (2014).

[17] Krstanoski, N.: Modelling Passenger Distribution on Metro Station Platform, *Inter. Journal for Traffic and Transport Engineering* (2014).

[18] Bosina, E. and et al.: Distribution of passengers on railway platforms (2015).

[19] Zhang, F. and et al: Spatio-Temporal Segmentation of Metro Trips Using Smart Card Data, *IEEE Trans. On vehicular Technology.* (2015).

[20] Mohssen, N. and et al.: It's the human that matters: accurate user orientation estimation for mobile computing applications, *MOBIQUITOUS* (2014).

[21] Chon, Y. and et al.: Understanding the coverage and scalability of place-centric crowdsensing, *UbiComp*, ACM (2013).

[22] Elhamshary, M. and et al.: TransitLabel: A Crowd-Sensing System for Automatic Labeling of Transit Stations Semantics, *MobiSys*, ACM (2016).

[23] Breiman, L.: Random forests, *Machine learning* (2001).

[24] Muralidharan, K. and et al.: Barometric phone sensors: more hype than hope!., *HotMobile*, ACM (2014).

[25] Hemminki, S. and et al.: Accelerometer-based transportation mode detection on smartphones, *SenSys*, ACM (2013).

[26] Sankaran k., a. e. a.: Using Mobile Phone Barometer for Low-Power Transportation Context Detection, *SenSys*, ACM (2014).

[27] Higuchi, T. and et al.: Tracking Motion Context of Railway Passengers by Fusion of Low-Power Sensors in Mobile Devices, *ISWC*, ACM (2015).

[28] Jelinek, F. and et al.: Design of a linguistic statistical decoder for the recognition of continuous speech, *IEEE Trans. on Information Theory* (1975).

[29] Scott, S.: Bayesian methods for hidden Markov models, *Journal of the American Statistical Association* (2011).

[30] Viterbi, A. J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *IEEE Trans. on Information Theory* (1967).

[31] Elhamshary, M. and et al.: Activity recognition of railway passengers by fusion of low-power sensors in phones, *SIGSPATIAL*, ACM (2015).

[32] Fan, M. and et al.: Public restroom detection on mobile phone via active probing, *ISWC*, ACM (2014).

[33] : Hitachi Echo-escalators., http://www.hitachi.com/environment/showcase/solution/industrial/escalator.html.

[34] : Mitsubishi Echo-elevators, http://www.mitsubishielevator.co.th/2016/company-detail.php?environment.

[35] : Escalator usage in transits., http://www.nytimes.com/2008/08/12/nyregion/12escalators.html?_r=0.

[36] Kullback, S. and Leibler, R. A.: On information and sufficiency, *The annals of mathematical statistics* (1951).

[37] Alzantot, M. and et al.: UPTIME: Ubiquitous pedestrian tracking using mobile phones, *WCNC*, IEEE (2012).

[38] Azizyan, M. and et al.: SurroundSense: mobile phone localization via ambience fingerprinting, *MobiCom*, ACM (2009).

[39] : F1 Score., https://en.wikipedia.org/wiki/F1_score.

[40] Gordon, M. and et al: PowerTutor: a power monitor for android-based mobile platforms (2013).

©2016 Information Processing Society of Japan