Electronic Preprint for Journal of Information Processing Vol.28

Regular Paper

Barrier Detection Using Sensor Data from Multiple Transportation Modes

Yuki Kurauchi[1,a)] Naoto Abe[1] Osamu Matsuda[1] Hitoshi Seshimo[1]

Received: November 6, 2019, Accepted: June 1, 2020

Abstract: Providing routes that are passable allowing movement by transportation modes such as wheelchairs or strollers requires accessibility information including details about the type and location of barriers. Earlier research detected the types of barriers using sensor data from people in wheelchairs and able-bodied people, but the amount and range of travel of people in wheelchairs is limited. Also, small barriers are not noticed by able-bodied persons since they tend to be easily ignored during movement. In our research, the goal was to detect barrier details using sensor data from various transportation modes. However, there are issues with the selection of barrier detection models for each mode and with the cost of collecting data. To overcome this model-selection problem, we propose a model that detects transportation modes and barriers in two stages. We also propose a method for reducing the cost of collecting data, with which we prepare a course with a smooth surface, collect data, and simulate rough surfaces by adding noise. We conducted three experiments to verify the effectiveness of the proposed method. The proposed method achieved an accuracy of 91.5% in detecting six transportation modes, and showed that adding noise increased accuracy by 3.7 percentage points on rough surfaces. When detecting eight types of barriers, our method achieved an accuracy of 87.7% for walking with a stroller, and showed that adding noise increased accuracy by 6.8 percentage points on rough surfaces. Therefore, the proposed method is effective in detecting barrier details using multiple transportation modes.

Keywords: sensing, transportation mode recognition, barrier detection, CNN, data augmentation

1. Introduction

For people using wheelchairs or strollers, barriers such as small steps can make it difficult to proceed, so in unfamiliar places, they are forced to find a way around such barriers. Automatically providing alternates routes to a destination that match the attributed of the transportation mode allows people to move with confidence even in places they have never been before. To achieve this, the location and type of barriers must be obtained beforehand. Different transportation modes can help persons move through different types of barriers depending on their attributes. For example, a wheelchair may not be able to pass a step of 2 cm but a stroller can. Therefore, details of barriers are needed such as the size of a step/projection or the gradient of a slope. There is earlier research on detecting the locations and types of barriers using sensor and GPS data from people in wheelchairs and those who are able-bodied [2], [3]. However, persons in wheelchairs have only a limited travel range and make only a limited number of trips so it is difficult to keep the latest barrier information updated over wide areas. Most able-bodied persons on the other hand can easily stride over small projections or steps so we assume such barriers will not be detected from their data. In this research, our goal was to detect barrier details using sensor data from various transportation modes, including walking, wheelchairs, walking with strollers and others, to guarantee the number of users (quantity of data) and types of barriers that can be detected (quality of data).

We assume that there are three main problems to be solved in detecting barriers using multiple transportation modes.

- Problem 1: The barrier detection model for each transportation mode must be selected.
- Problem 2: Tuning of these models for each transportation mode is difficult.
- Problem 3: A huge amount of training data is required.

Below, we discuss details of these problems and how we solved them in this research. The first problem is the need to select a barrier detection model particular to the transportation mode, assuming that creating separate barrier detection models for each transportation mode will be more accurate. It would be possible to have the participants set their current transportation mode, but this would be inconvenient and subject to issues such as participants forgetting to set the mode. Therefore, we propose a two-stage model in which we first detect the transportation mode from the sensor data then use a barrier-detection model based on the result. For the second problem, we assume that higher detection accuracy can be obtained by tuning the barrier-detection model and parameters being used for each transportation mode. However, it is not practical to find an optimal model for each mode. If for example, six transportation modes are used, six models would need to be tuned separately. In this research, we trained the same shape model with data for each transportation mode and used the

[1] NTT Service Evolution Laboratories, Hikarinooka, Yokosuka, Kanagawa 239–0847, Japan
[a)] yuuki.kurauchi.mv@hco.ntt.co.jp

This paper is an extended version of our COMPSAC'19 conference paper [1].

© 2020 Information Processing Society of Japan

instances to detect the modes and barriers. For the third problem, training data for each transportation mode are needed, and to detect barriers regardless of the path surface (e.g., a rough asphalt path), sensor data for various types of surfaces and barriers are needed thus increasing the cost of data collection. Thus, we also propose a method for collecting data on routes with a smooth surface and then adding noise to these data to simulate rough surfaces. This eliminates the need to collect data for rough surfaces and reduces data-collection costs, while enabling detection of barriers regardless of the path surface.

We conducted three experiments to verify the effectiveness of the proposed method and achieved an accuracy of 91.5% in detecting six transportation modes, and adding noise increased accuracy by 3.7 percentage points for rough surfaces. When detecting eight types of barriers for walking with a stroller, we achieved an accuracy of 87.7% and showed that adding noise increased accuracy by 6.8 percentage points for rough surfaces. This showed the effectiveness of the proposed method and enabled detailed barrier detection using multiple transportation modes.

Contributions of this research are as follows:

- We propose a two-step model, in which we first detect the transportation mode from the sensor data then detect the types of barrier using a barrier-detection model specific to that transportation mode.
- We evaluated the accuracy for detecting six transportation modes (walking, using a wheelchair, walking with a stroller, walking with a handcart, walking with a suitcase, and using a bicycle).
- We evaluated the accuracy of detecting barriers using the same shape model trained with data from each transportation mode.
- We evaluated the accuracy in detecting eight types of barriers (level, 2, 8, and 16 cm steps, 5% and 10% slopes, stairs, and tactile paving) and show high accuracy in practical terms.
- We propose a method for collecting data on courses with smooth surfaces then adding noise simulating a rough surface.
- We confirm that the accuracy in detecting transportation modes and barriers on rough surfaces can be improved using the proposed method.
- We solve the problems of model tuning and training data collection and show that barrier sensing using multiple transportation modes is practical.

Document organization: Section 2 discusses related research. Section 3 describes the proposed methods. Section 4 describes the experiments. Section 5 discusses the results. Section 6 gives conclusions and future prospects.

2. Related Research

Sensing has been used to detect user actions [4], [5] and various other behaviors [6]. Some research has been done on sensing roadway surfaces [7], but our interest is in walking routes. In particular, crowd sensing refers to detection methods using information from many users and is widely used for tasks such as estimating levels of congestion in train stations [8] or creating detailed maps [9]. Our method could be called a type of crowd sensing for barrier acquisition.

In barrier-acquisition research, there has been a study on facilitating surveys manually by volunteers [10] or residents [11]. Surveys can include aspects that are difficult to acquire using sensing, such as whether toilets accommodate persons with disabilities or the widths of walking routes. Therefore, it is necessary to use both high-cost survey methods that can collect detailed aspects, and low-cost methods using sensors, which can collect only a few aspects. There was another study on acquiring barriers from Google Street View [12]. They collected barrier information on a large scale at low cost and used the data to create a web service, but detecting small barriers is difficult due to resolution and occlusion problems. Also, because this study depends on the update frequency of Google Street View data, this study might not be able to collect the latest barrier details. We assume renewal frequency is an important factor because barriers are constantly changing due to road work and other reasons.

Regarding barrier acquisition using crowd sensing, there was a study on detecting barriers using sensor data from wheelchairs [2]. Because there are relatively few wheelchair users and their movements are limited, others have conducted barrier detection using sensors with able-bodied people [3]. They are also improving their collection method through a gamification approach [13]. Able-bodied persons however are able to easily stride over small steps/projections while walking so we assume that such barriers cannot be detected from their data. Nonetheless, these results can help improve detection accuracy for some of the transportation modes examined in our research and can be used together with our proposed method. These prior methods included detecting slopes and steps, but our research expands this by also detecting the degree of slopes and height of steps with finer granularity.

Other methods have been used to detect barriers, such as attaching specialized sensors to shoes [14] or to the wheels of a wheelchair [15], but we chose a method that can be implemented at low cost using the user's smartphone. Video from in-vehicle cameras has also been used to estimate the road-surface conditions [16], but these methods are difficult to apply to walking route surfaces because vehicles driving on the road are some distance from the walking routes, and walking routes are often obstructed by guard rails or other objects.

Some studies focused on detecting transportation modes using GPS [17] and sensors [18], but these dealt with walking, running, cars, buses, trains, motorbikes, and bicycles, and no with other modes such as wheelchairs, strollers, handcarts, or suitcases.

Data augmentation is widely used in many machine learning tasks such as image recognition [19] and translation [20]. Sensor-data augmentation has been widely studied as a means to detect symptoms of Parkinson's disease [21]. Our proposed method of adding noise is a type of jittering, but is different from other methods in that rather than to increase accuracy, we use it to reduce the cost of data collection. It has been reported that jittering is not effective in increasing the accuracy of detecting symptoms, but for this research we hypothesized that it would be effective for detecting transportation modes and barriers on different types of surfaces.

© 2020 Information Processing Society of Japan

There have been studies that used sensing to conserve power and communication and reduce latency. Compressed Learning [22] makes observations in a way that reduces the number of samples required for detection, and there has been research on conserving communication by computing the earlier parts of the deep neural network (DNN) model at the sensors then transmitting the results from the intermediate layers, where there are fewer nodes, to the server to compute the remainder of the model [23]. In particular, when using a user's smartphone for crowd sensing, conserving power and volume of communication are major issues, so application of these methods should reduce the burden on users and make it easier for them to work with our method.

3. Proposed Method

The objective of this research was to collect information regarding the types of barriers beforehand to automatically present routes that are traversable according to users' attributes. By attributes, we mean aided or unaided wheelchairs, strollers, walking sticks and visual disabilities. We extend the "Development Specification for Spatial Network Model for Pedestrians," from the Ministry of Land, Infrastructure and Transport and Tourism of Japan [24] by adding definitions for barriers though which the user can move (**Table 1**). Although those with visual disabilities can traverse any barriers that people without disabilities can, information regarding tactile paving is also needed to present traversable routes for such individuals. Therefore, we need to detect a total of seven types of barriers, i.e., 2, 8, and 16 cm steps; 5% and 10% slopes; stairs; and tactile paving. Tactile paving is not strictly speaking a barrier, but for simplicity the references to barriers in this article also include it.

In this research, we used data from sensors in smartphones to detect barriers. We used five types of sensors: three-axis acceleration, three-axis gyroscope, three-axis gravity, three-axis magnetism, and air pressure; totaling 13 dimensions. To ensure an adequate number of users (quantity) and that different types of barriers can be detected (quality), we attempted to detect barrier details using sensor data from various transportation modes. The six transportation modes we used were walking, wheelchair, stroller, handcart, suitcase, and bicycle (**Figs. 1** and **2**). Suitcase refers to the type with wheels and a handle. We selected those modes in addition to wheelchair since they are routinely used by most able-bodied persons on the walkway.

Table 1 Definition of passable barriers by attribute.

Barrier type		Wheelchair Unaided	Wheelchair aided	Stroller	Walking stick	Able-bodied
Slope	up to 5%	Y	Y	Y	Y	Y
	up to 10%	N	Y	Y	Y	Y
	10% or more	N	N	N	Y	Y
Step	up to 2 cm	Y	Y	Y	Y	Y
	up to 8 cm	N	Y	Y	Y	Y
	up to 16 cm	N	N	N	Y	Y
	16 cm or more	N	N	N	N	Y
Stairs		N	N	N	N	Y

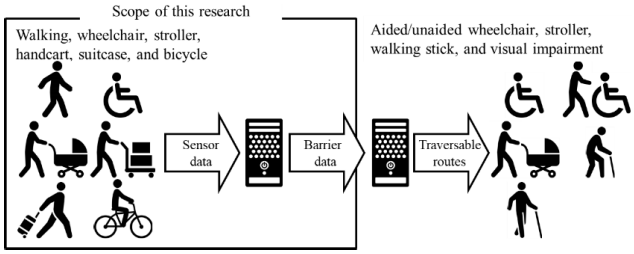


Fig. 2 Scope of this research. We collect information regarding types of barriers beforehand to automatically present routes that are usable according to user attributes.

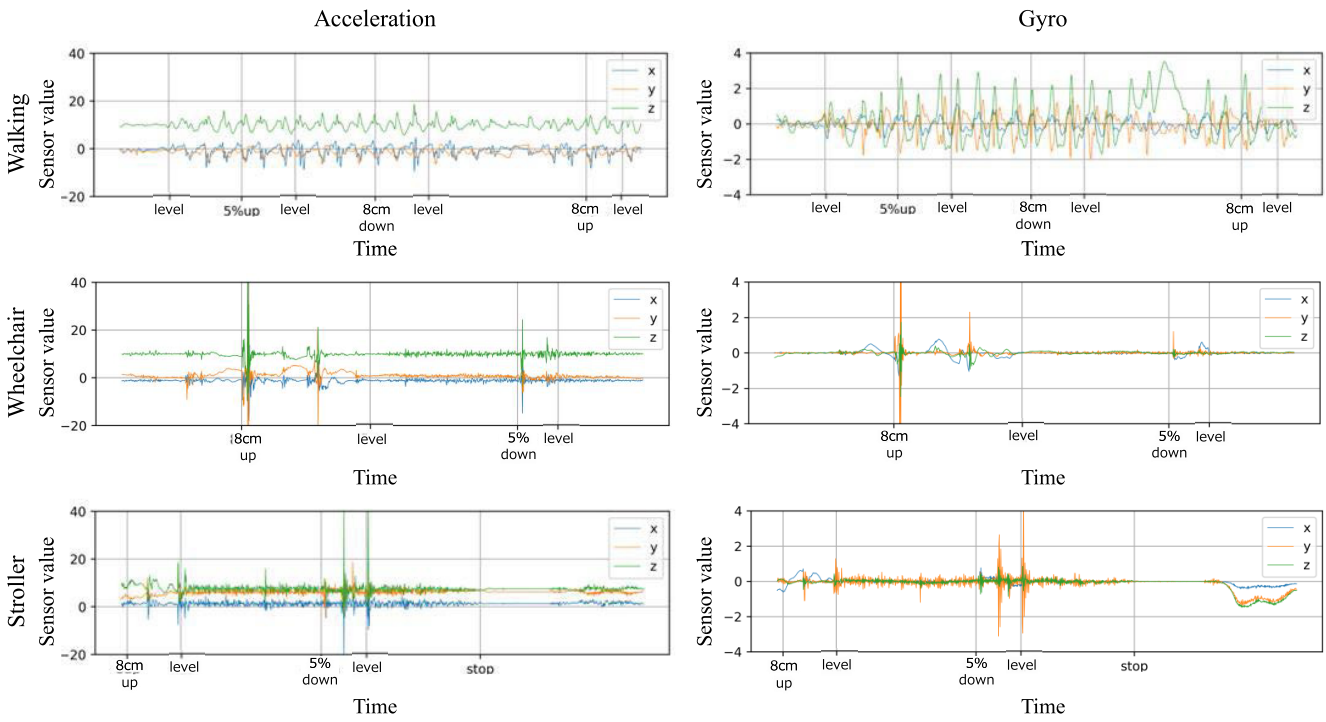


Fig. 1 Acceleration and gyro data on smooth surface (15 seconds). Horizontal axis indicates labeled barriers. Walking shows stepwise pattern and stroller data are noisier than those of wheelchair.

© 2020 Information Processing Society of Japan

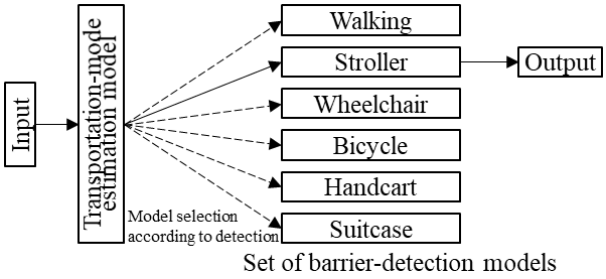


Fig. 3 Proposed two-step detection model for transportation modes and barriers. Example of stroller as estimated transportation mode.

Table 2 Features used and number of dimensions.

Feature	No. of dimensions
Mean on each axis	13
Standard deviation on each axis	13
Mean root sum of squares	5
Distance between peaks	13
Histogram	130
Power spectrum	1,300
Power spectrum histogram	130
Frequency component entropy	5
Covariance between axes	13
Autocorrelation coefficients	130

Detecting barriers with multiple transportation modes presents three problems.

- Problem 1: The barrier detection model for each transportation mode must be selected.
- Problem 2: Tuning of models for each mode is difficult.
- Problem 3: A huge amount of training data is required.

Below, we discuss these problems and how we solved them.

For Problem 1, some modes show stepwise patterns and some shows noisy data and therefore we assume that the characteristics of the sensor data will differ for different transportation modes (Fig. 1) and that having separate barrier-detection models for each transportation mode will yield higher detection accuracy. In that case, the barrier-detection model to be used must be selected for each detection. We could require the user to set the transportation mode currently being used each time, but this would be troublesome, and users could forget to set it properly. Therefore, our two-step detection model first detects the transportation mode from the sensor data then uses the barrier-detection model corresponding to the detected transportation mode (**Fig. 3**).

For Problem 2, we assume that by changing the models and parameters used for each transportation mode, we will be able to achieve higher detection accuracy, but it is not practical to find optimal models for each transportation mode. Given that we have six transportation modes, each of these models would need to be tuned separately. In this research, we reduced the time required for tuning by training the same shape model separately with data from each transportation mode and using the instances to detect these modes and barriers. We use the features listed in **Table 2** for detection. The barrier and mode detection model we used was a convolutional neural network (CNN) consisting of three convolutional layers and two fully connected layers, with sensor data and power spectrum as the input to the convolutional layers, and features as the input to the fully connected layers (**Fig. 4**). We set the parameters to be the same as those used by Yang et al. [5].

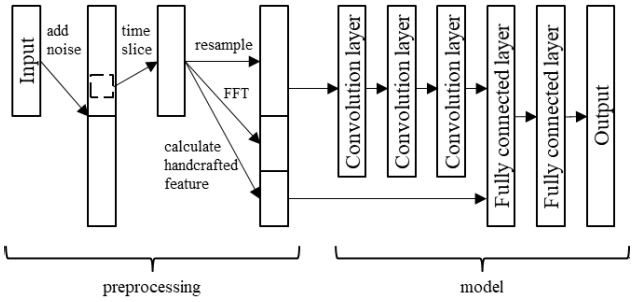


Fig. 4 Input data preprocessing and model overview.

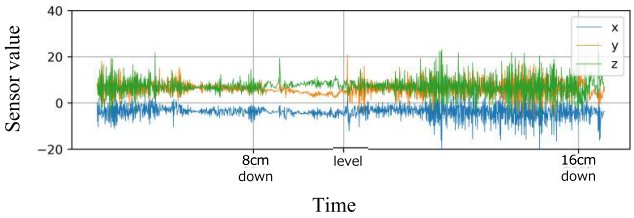


Fig. 5 Acceleration data for rough surface with stroller (15seconds). Horizontal axis indicates labeled barriers. Noise is added by rough surface.

Problem 3 deals with the high cost of collecting data since training data are needed for each transportation mode, and we need to be able to detect barriers regardless of the qualities of surfaces such as rough asphalt. Trends in sensor data differ greatly depending on the type of surface being traversed (**Fig. 5**), and sensor data for traversing various types of barriers with various types of surfaces are needed. Also, if we want to collect data by walking around the town freely, we need to apply highly accurate labels to the data, which requires measurements of locations to create the labels. Huge amounts of data must also be acquired to obtain data for types of barriers that only appear infrequently. This can result in extremely high costs. Therefore, our proposed method involves using a smooth course and simulated rough surfaces by adding noise to data acquired on the smooth course. Preparing a course enables us to deal with barriers that only appear infrequently in a town. This is equivalent to what would be called a phoneme balanced corpus in speech research, which contains all phonemes within a small corpus rather than using a larger general corpus [25]. By adding noise to simulate rough surfaces, data need not be collected for that type of rough surface. This reduces data-collection costs to a fraction of the type required for the road surface to be handled. For example, when dealing with smooth roads and rough asphalt, the cost is reduced by a factor of two. We used white Gaussian noise which includes all frequencies, to ensure it would be robust with various types of path noise. Values were chosen such that the mean was zero and the variance was the difference in variance between moving on a smooth or rough surface, as shown in **Table 3**. We also ensured that the variance of the noise was equal on all three axes, so that detection would not depend on the orientation of the smartphone. An example of sensor data from a smooth surface with noise added is shown in **Fig. 6**. Both data with noise added and without noise added were used as training data, so the amount of training data was twice that of the input data.

© 2020 Information Processing Society of Japan

Table 3 Difference in variance of normalized sensor values due to path surface.

Transportation mode	Acceleration	Gyro	Gravity	Magnetism	Air pressure
Walking	0.12	0.25	0.29	0.10	0.15
Wheelchair	0.06	0.11	0.02	0.21	0.00
Stroller	0.37	0.16	0.12	0.15	0.08
Handcart	0.45	0.40	0.17	0.15	0.02
Suitcase	0.13	0.14	0.06	0.12	0.05
Bicycle	0.08	0.04	0.10	0.20	0.05

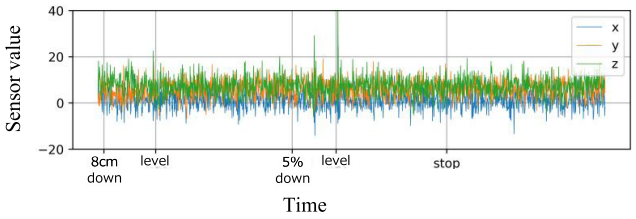


Fig. 6 Stroller data from Fig. 1 (bottom-left) with artificial noise added. Horizontal axis indicates labeled barriers.

4. Experiments

We conducted three experiments to verify the effectiveness of the proposed method. Below, we describe the course used in the experiments and how the data were collected and adjusted then give an overview of the three experiments.

4.1 Course Conditions

For smooth surfaces, we constructed an indoor course out of wood for the barriers to be detected (**Fig. 7**). For rough surfaces, we found locations outside for each of the barrier types. The surfaces in the outdoor locations were asphalt. None of the outdoor courses included tactile paving. When courses included multiple barriers, we ensured that they were separated by an ample distance of 2 m or more. Courses with tactile paving were prepared with linear directional blocks in the direction perpendicular to traveling direction.

4.2 Data Gathering Conditions

We collected sensor data while traversing the above courses three times in both directions and collected video, which was used to apply accurate labels to the data. A total of 28 participants took part in our smooth-course trials: two males and two females of ages in each 10 year period from 10 to 80. For the trials with rough surfaces, one female and one male participated from each of the same 10 year periods, except for a female in her 70 s: totaling 13 in instead of 14 participants. The participants for the indoor and outdoor experiments were different groups with no overlap. While walking, smartphones were held in the usual places such as in a shirt or trouser pocket or in a handbag. For the bicycle, the phone was placed under the seat, and in the remaining cases, it was attached near the handle of the mode being used (**Fig. 8**). While detecting, we assumed that the location of the smartphone while walking was unknown. Data for stairs was only obtained for the cases of walking and suitcase, and no data were collected for cases in which moving through the barrier was impossible (wheelchair fo 16 cm steps/projections and bicycle for 8 cm steps/projections or larger). The stroller and suitcase

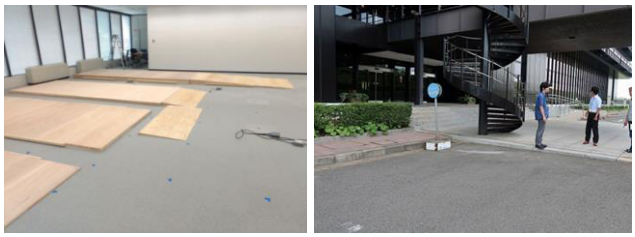


Fig. 7 Wooden course for collecting smooth-path data (left) and outside course for collecting rough-path data (right).

Fig. 8 Places where sensor and weight are attached.

Table 4 Sensors used and sampling rates.

Sensor type	Rate [Hz]
Acceleration	400
Gyro	400
Gravity	200
Magnetism	100
Air pressure	32

Table 5 # of collected samples.

	Level	2cm	8cm	16cm	5%	10%	Tactile	Stair	**Total**
Walking	43,125	1,909	1,817	1,843	8,194	9,648	1,359	11,966	**79,861**
Wheelchair	58,969	1,346	4,913	0	10,106	3,589	695	0	**79,618**
Stroller	56,263	2,639	2,692	3,958	8,718	11,199	685	0	**86,154**
Handcart	54,819	2,863	2,638	4,067	8,290	10,536	821	0	**84,034**
Suitcase	60,698	1,229	1,141	1,799	7,288	9,480	354	13,897	**95,886**
Bicycle	48,359	748	852	0	8,029	3,822	951	0	**62,761**
Total	**322,233**	**10,734**	**14,053**	**11,667**	**50,625**	**48,274**	**4,865**	**25,863**	**488,314**

each weighed 4 kg, and the wheelchair weighed 50 kg. Xperia XZ smartphones with Android 8.0 were used to collect the data. The highest sampling rates possible for each sensor were used as shown in **Table 4**. This resulted in 15.9 hours of data. Data were then resampled at 400 Hz, a fast Fourier transform (FFT) was applied for 1,500 ms time windows moving the window with 100 ms steps, then resampled at 20 Hz before use (Fig. 4, **Table 5**).

4.3 Data Tuning

The data from each sensor were standardized. For sensors with three axes, the mean and variance using values from all three axes were used.

To confirm the orientation of the smartphone, rotations of angles θ and φ were computed that satisfy (1) below using the gravity sensor, and the rotations were applied to the data from each sensor. This means the orientation of the smartphone in relation to gravity was uniform.

© 2020 Information Processing Society of Japan

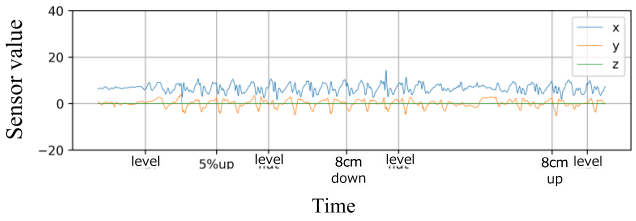


Fig. 9 Results of rotation applied to walking data in Fig. 1 (top-left): x: forward, y: vertical, z: left and right. Horizontal axis indicates labeled barriers.

$$
\begin{aligned}
& rR_z(\theta)R_y(\phi)\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\\ \mathbf{z}\end{pmatrix} \\
&= r\begin{pmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}\cos\phi & 0 & \sin\phi\\ 0 & 1 & 0\\ -\sin\phi & 0 & \cos\phi\end{pmatrix}\begin{pmatrix}x\\ y\\ z\end{pmatrix} \\
&= r\begin{pmatrix}\cos\theta\cos\phi & -\sin\theta & \cos\theta\sin\phi\\ \sin\theta\cos\phi & \cos\theta & \sin\theta\sin\phi\\ -\sin\phi & 0 & \cos\phi\end{pmatrix}\begin{pmatrix}x\\ y\\ z\end{pmatrix} \\
&= r\begin{pmatrix}x\cos\theta\cos\phi - y\sin\theta + z\cos\theta\sin\phi\\ x\sin\theta\cos\phi + y\cos\theta + z\sin\theta\sin\phi\\ -z\sin\phi + z\cos\phi\end{pmatrix} = \begin{pmatrix}1\\ 0\\ 0\end{pmatrix} \qquad (1)
\end{aligned}
$$

where r is a coefficient largely consistent with the reciprocal of the acceleration of gravity when at rest and R_z and R_y represent rotations in the directions of the z and y axes, respectively, θ and φ are the rotation angles in the directions of the z and y axes, respectively, and $(x\ y\ z)^T$ are the sensor values for the x, y, and z axes. As a result, the components in the left and right direction were near zero (**Fig. 9**).

Considering that there could be some error in the manually applied labels, we applied automatic correction using properties of the data in the segments between neighboring labels. Specifically, we computed the mean and standard deviation of sensor data on segments with labels assigned and moved labels to maximize the difference in absolute values of the mean and standard deviation between neighboring segments as in Eq. (2). This is based on the assumption that the properties of the sensor data differ in each labeled section. We correct errors due to manual labeling by maximizing the difference between adjacent sections and prevent incorrect learning caused by incorrect labels.

$$
\begin{aligned}
t_{n_{new}} &= t_n + \underset{-s\le\Delta t\le s}{\operatorname{argmax}}(f(a_{t_{n-1}:t_n+\Delta t}, a_{t_n+\Delta t:t_{n+1}}) \\
&\quad + wg(a_{t_{n-1}:t_n+\Delta t}, a_{t_n+\Delta t:t_{n+1}})) \\
f(a_1, a_2) &= |\mathrm{avg}(a_1) - \mathrm{avg}(a_2)| \\
g(a_1, a_2) &= |\mathrm{std}(a_1) - \mathrm{std}(a_2)|
\end{aligned} \qquad (2)
$$

where t_n is the time of the nth label, $a_{t_{n-1}:t_n+\Delta t}$ represents the sensor data on the segment from the $n-1$th label to nth label, shifted by Δt. Here, s is the range over which labels can be shifted, which we set to 100 ms. Also, w is a weighting, which we set to 1, avg indicates the mean, and std indicates the standard deviation.

4.4 Experimental Conditions

A total of 20 epochs were performed and the weightings of the loss functions for each class were the inverse of the frequency of appearance in the training data. To prevent overfitting, we applied L2 normalization and dropout to half of the nodes of the fully connected layers. We used a batch size of 512, and ReLU as the activation function. We used Adam as the optimizer and optimized the parameters using hyperopt [26]. We used chainer for implementation.

We used cross-validation, but rather than using random partitions, we used a user's data as test data and data from the remaining users as training data. We used accuracy, precision, recall, and *F-measure* (F_1) as the performance measure. When there were multiple classes, we used index weighted by the number of data points per class with prefix "w-", such as in Eq. (3).

$$
\text{w-}F_1 = \sum_i 2 * \omega_i \frac{precision_i \cdot recall_i}{precision_i + recall_i} \qquad (3)
$$

where i is the class index, ω_i is the weighting for class i, computed as n_i/N, where n_i is the number of data points in class i, and N is the total number of data points. $precision_i$ is the precision for class i, and $recall_i$ is the recall for class i.

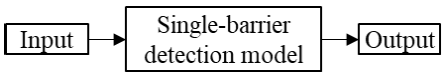


Fig. 10 Simple end-to-end model as comparison model. There is no transportation-mode detection step.

4.5 Experiment 1: Transportation-mode Detection

The objectives of Experiment 1 were to determine whether it is possible to detect the transportation mode using the proposed method, and whether it is possible to detect a mode over rough surfaces by adding noise to the data obtained using smooth surfaces. To do so, we computed the accuracy for detecting a transportation mode in the cases of using data from the smooth surface and data from the smooth surface with noise added as training data, and using data from a smooth surface and data from a rough surface as the test data.

4.6 Experiment 2: Barrier Detection

The objectives of Experiment 2 were to verify the results of detecting barrier details using multiple transportation modes, whether the barriers could be detected using the proposed two step model, and whether barriers on paths with rough surfaces could be detected by adding noise to the data obtained using smooth surfaces. To achieve these objectives, we compared the accuracy of total individual models trained with data for each transportation mode (Fig. 3), with that of a simple end-to-end model trained using data from all transportation modes (**Fig. 10**) (hereafter, our proposed model is referred to as the total model and the other as the simple model). We also computed the detection accuracy for each barrier-detection model of a transportation mode and class of barrier type, for cases using data from smooth paths with and without noise as the training data and data from smooth paths and rough paths as the test data.

4.7 Experiment 3: VS Conventional Augmentation Method

The objectives of Experiment 3 were to verify the results from

© 2020 Information Processing Society of Japan

comparing the proposed method with a conventional augmentation method. In the barrier detection task, we compared the proposed method, which augments a rough road surface from a smooth road surface without reducing the number of participants, and a conventional method of augmenting them using both rough and smooth road surfaces. Since the compression ratio of the proposed method is fixed at one section of the type of road surface to be handled, we calculated barrier-detection accuracy by changing the number of subjects used for learning the comparison method.

5. Results and Discussion

We discuss the results of the experiments described in the previous section. Below, we refer to sets of training data and test data as <training data> → <test data>. For example, the case using data from smooth surfaces for training data and data from rough surfaces for testing is be referred to as "Smooth → Rough".

5.1 Experiment 1: Transportation Mod Detection

The results from detecting each class of transportation mode in the Smooth → Smooth case are shown in **Table 6**. Transportation mode detection accuracy was high at 91.5% with weighted F_1 of 0.914. The F_1 values for individual transportation mode classes show that bicycle class was the highest, with 0.962. This high detection accuracy could be due to several characteristics such as the slow waves observed from the gyro (**Fig. 11**, upper). Most of the other classes yielded F_1 scores over 0.9, but the walking class was lowest, with 0.866, and suitcase class was the next lowest, with 0.905. These results were low because the recall for walking was 0.806, and the precision for suitcase was 0.868.

The results for each combination of training and test data are shown in **Table 7**. The Smooth → Rough case resulted in a lower accuracy of 87.4% than the Smooth → Smooth case with F_1 of 0.874, indicating that the different surface made detection more difficult. However, the Noise-added → Rough case produced an accuracy of 91.1% and F_1 of 0.912, with 3.7 percentage points higher accuracy and higher F_1 by 0.038 than the Smooth → Rough case. The Noise-added → Smooth case also resulted in an accuracy of 92.1% and F_1 of 0.920, comparable to the Smooth → Smooth case, indicating that adding the noise produced no negative effect. Note also that accuracy for the Rough → Rough case was 84.9% and F_1 was 0.840, which is lower than those for the Noise-added → Rough case using the proposed method. This could be a result of the generalizing effect of data augmentation, which is one of the original purposes of the data augmentation discussed by Lorintiu et al. [20].

The confusion matrix for the Smooth → Smooth case (**Table 8**, upper-left) shows that walking was most often misdetected as suitcase, and wheelchair was most often misdetected as handcart. These errors may be due to the acceleration repeated with each step (Fig. 11 middle and bottom) in the former case, and that both of the latter have four wheels and are of similar size. The confusion matrix for the Smooth → Rough case (Table 8, lower-left) shows that misdetection of walking as suitcase accounted for most errors, but in the confusion matrix for Noise-added → Rough (Table 8, upper-right) misdetection of walking as suitcase decreased drastically.

Table 6 Experiment 1 (transportation-mode detection) results for Smooth → Smooth by class of transportation mode. Maximum values for each class of transportation mode are shown in bold.

Class	Accuracy	Precision	Recall	F_1
Walking	-	0.935	0.806	0.866
Wheelchair	-	0.932	0.900	0.915
Stroller	-	0.924	**0.981**	0.952
Handcart	-	0.896	0.951	0.923
Suitcase	-	0.868	0.945	0.905
Bicycle	-	**0.949**	0.975	**0.962**
Total	91.5%	0.916	0.915	0.914

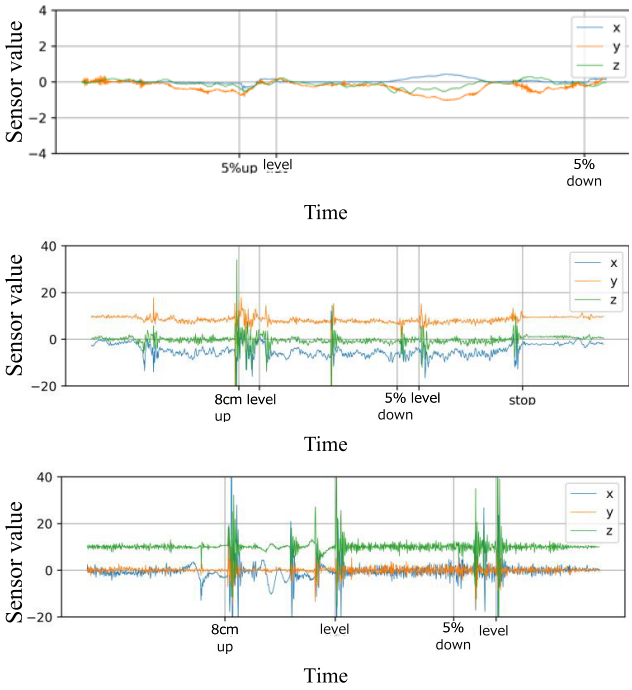


Fig. 11 Collected data for smooth surface (15 seconds). Examples for gyro of bicycle (top) acceleration of suitcase (middle) and handcart (bottom). Suitcase shows a stepwise pattern in x direction. Horizontal axis indicates labeled barriers.

Table 7 Experiment 1 (transportation-mode detection) results for each combination of training and test data.

Train	Smooth		Noise added		Ref.: Rough
Test	Smooth	Rough	Smooth	Rough	Rough
Accuracy	91.5%	87.4%	92.1%	91.1%	84.9%
w-precision	0.916	0.878	0.924	0.912	0.846
w-recall	0.915	0.874	0.921	0.911	0.849
w-F_1	0.914	0.874	0.920	0.912	0.840

5.2 Experiment 2: Barrier Detection

The results from each barrier-detection model of a transportation mode in the Smooth → Smooth case ares shown in **Table 9**. For the simple model, trained with data from all transportation modes, the accuracy was 79.7% and weighted F_1 was 0.742, but the total of results from all models trained with data from each transportation mode yielded an accuracy of 81.0% and F_1 of 0.779. Thus, training for each transportation mode produced an increase in accuracy of 1.3 percentage points, and 0.037 in the weighted F_1 value. The level class makes up a large fraction of barriers so accuracy did not increase much, but the weighted F_1 increased significantly, and this is assumed that accuracy increased for the classes other than level class.

When examining individual barrier-detection models of transportation mode for the proposed method, the bicycle model had

© 2020 Information Processing Society of Japan

Table 8 Confusion matrices for experiment 1 (transportation-mode detection).

Smooth→Smooth

Actual label \ Estimated label	Walking	Wheelchair	Stroller	Handcart	Suitcase	Bicycle	**Total**
Walking	79,948	2,329	4,382	1,295	9,885	1,339	**99,178**
Wheelchair	485	61,195	422	5,559	182	177	**68,020**
Stroller	741	95	66,060	25	148	237	**67,306**
Handcart	518	1,895	261	59,619	239	130	**62,662**
Suitcase	3,235	134	219	52	70,861	459	**74,960**
Bicycle	570	27	121	23	368	43,507	**44,616**
Total	**85,497**	**65,675**	**71,465**	**66,573**	**81,683**	**45,849**	**416,742**

Noise added→Rough

Actual label \ Estimated label	Walking	Wheelchair	Stroller	Handcart	Suitcase	Bicycle	**Total**
Walking	22,457	596	673	5	1,928	8	**25,667**
Wheelchair	5	16,323	0	2,840	129	0	**19,297**
Stroller	1,305	40	27,347	4	9	23	**28,728**
Handcart	2	2,287	7	25,333	120	0	**27,749**
Suitcase	2,321	0	0	0	28,929	5	**31,255**
Bicycle	486	1	10	0	39	11,451	**11,987**
Total	**26,576**	**19,247**	**28,037**	**28,182**	**31,154**	**11,487**	**144,683**

Smooth→Rough

Actual label \ Estimated label	Walking	Wheelchair	Stroller	Handcart	Suitcase	Bicycle	**Total**
Walking	22,704	866	324	238	7,335	96	**31,563**
Wheelchair	142	18,343	8	2,551	0	0	**21,044**
Stroller	289	39	28,381	103	1,033	51	**29,896**
Handcart	97	2,634	63	26,203	79	0	**29,076**
Suitcase	2,764	34	0	51	29,634	1	**32,484**
Bicycle	784	39	4	9	104	11,599	**12,539**
Total	**26,780**	**21,955**	**28,780**	**29,155**	**38,185**	**11,747**	**156,602**

100,000
10,000
1,000
100
10
0

Table 9 Experiment 2 (barrier detection) results for Smooth → Smooth by model of transportation mode. Maximum values for each model in bold.

Model	**Accuracy**	**w-precision**	**w-recall**	**w-F_1**
Simple	79.7%	0.761	0.797	0.742
Total	81.0%	0.777	0.810	0.779
Walking	60.2%	0.498	0.602	0.536
Wheelchair	84.2%	0.803	0.842	0.806
Stroller	87.7%	0.859	0.877	**0.864**
Handcart	80.7%	0.758	0.807	0.765
Suitcase	84.6%	0.779	0.846	0.793
Bicycle	87.8%	**0.873**	**0.878**	0.856

Table 10 Experiment 2 (barrier detection) results for Smooth → Smooth by class of barrier. Evaluation index is F_1. Maximum values for each class of barrier shown in bold. Hyphen indicates non-traversable barrier.

Class	**Simple**	**Total**	**Walk-ing**	**Wheel-chair**	**Stroller**	**Hand-cart**	**Suit-case**	**Bicycle**
Level	0.888	0.901	0.757	0.922	**0.953**	0.902	0.912	0.941
2 cm	0.025	0.091	0.000	0.039	0.137	**0.219**	0.007	0.074
8 cm	0.333	0.525	0.002	0.638	**0.653**	0.510	0.032	-
16 cm	0.523	0.632	0.060	-	**0.814**	0.676	0.348	-
5%	0.132	0.277	0.041	0.193	0.376	0.186	0.003	**0.595**
10%	0.223	0.350	0.112	0.445	**0.634**	0.173	0.000	0.529
Stair	0.675	0.669	0.397	-	-	-	**0.864**	-
Tactile	0.252	0.328	0.000	0.115	**0.756**	0.559	0.147	0.006
Total	0.742	0.779	0.536	0.806	0.864	0.765	0.793	0.856

the highest weighted precision at 0.873, and weighted recall at 0.878, and the second highest for weighted F_1 at 0.856. The stroller model had the second highest weighted precision at 0.859, and weighted recall at 0.877, and the highest weighted F_1 at 0.864. We assume this was because the stroller has four wheels, the slope angles were reflected directly in the data, and its wheels have less suspension than those of some of the other modes, and vibrations are reflected more directly, which increased the accuracy in small barriers. The barrier-detection model for walking had the lowest values for all indices, but as discussed below, it is useful for estimating some of the barriers.

The F_1 for detecting each class of type of barrier for each barrier-detection model of a transportation mode in the Smooth → Smooth case is shown in **Table 10**. The total model achieved a higher F_1 than the simple model in all of classes. The table also shows that for each transportation mode, there are barriers that are more easily or less easily detected, suggesting that barrier details could be detected using multiple transportation modes. For example, handcart achieved the highest F_1 of 0.219 for 2 cm, suitcase achieved the highest F_1 of 0.864 for stair, and bicycle achieved the highest F_1 of 0.595 for 5%. Walking showed a low F_1 of 0.000 for 2 cm and tactile paving. We consider that this is because small barriers are absorbed when walking. With suitcase, the user's arm absorbs the angle of a slope, so it is difficult to detect the degree of a slope, and for steps, a suitcase is usually lifted for steps of 8 or 16 cm, so it is difficult to detect the size of these steps. Stairs are only traversable when walking or with a suitcase, so these transportation modes can be used to detect them complementarily. In other words, walking showed a low F_1 of 0.397 but there are many participants, while suitcase achieved a high F_1 of 0.864 but there are fewer participants. Note that F_1 was low for the 2 cm steps but for practical use this F_1 could be manually increased by increasing the per-class weighting for 2 cm steps as described at the beginning of Section 4.4.

The accuracy and F_1 for sets of training and test data are given in **Table 11**. When examining changes in accuracy by adding noise, comparing the Smooth → Rough and Noise-added → Rough cases, showed that the accuracy of some transportation nodes improved and some degraded. Stroller improved the most, with accuracy increasing 6.8 percentage points from 41.9% to 48.7% and F_1 increasing by 0.047 from 0.469 to 0.516. Accuracy increased for handcart, walking, and bicycle, and decreased for wheelchair and suitcase. The F_1 increased for all modes except suitcase. We consider that this is because for wheelchair and bicycle, the wheels and suspension absorb vibration so the error due to the path surface was small, while stroller and handcart do not absorb vibration, so the effect of adding noise was greater. This tendency is also clear in Table 3, with accuracy increasing for transportation modes showing a larger change in variance when the noise is greater.

The confusion matrix for the simple model is at the top-left in **Table 12**, while for the total model is at the top-right. Comparing the number of detection label outputs, all the barriers showed increases. For step and slope, in particular, the outputs increased by approximately two times. Looking at error trends, many of the errors were from estimating a barrier was level class, but this

© 2020 Information Processing Society of Japan

is because the transportation mode was incorrectly estimated to be walking as discussed below. The most common errors were interpreting level as slope and between different grades of slope.

Table 11 Experiment 2 (barrier detection) results from barrier- detection model of transportation mode for each combination of training and test data. Maximum values for each train → test pair are in bold.

Index: Accuracy

Train	Smooth		Noise added		Ref.: Rough
Test	Smooth	Rough	Smooth	Rough	Rough
Walking	60.2%	38.5%	62.6%	39.6%	38.4%
Wheelchair	84.2%	**57.2%**	84.5%	53.5%	**78.3%**
Stroller	87.7%	41.9%	87.2%	48.7%	64.5%
Handcart	80.7%	36.4%	80.3%	42.9%	49.3%
Suitcase	84.6%	51.8%	84.5%	51.7%	58.5%
Bicycle	**87.8%**	54.8%	**87.6%**	**57.8%**	76.5%

Index: w-F_1

Train	Smooth		Noise added		Ref.: Rough
Test	Smooth	Rough	Smooth	Rough	Rough
Walking	0.536	0.304	0.544	0.314	0.367
Wheelchair	0.806	**0.535**	0.822	**0.576**	**0.783**
Stroller	**0.864**	0.469	**0.856**	0.516	0.636
Handcart	0.765	0.355	0.760	0.358	0.467
Suitcase	0.793	0.400	0.794	0.400	0.573
Bicycle	0.856	0.494	0.851	0.505	0.766

However, the error rate for distinguishing between steps and between slopes decreased. For example, with the simple model, a 5% slope was mistaken for a 10% slope in 1,351 of 5,704 instances (23.7%) and 2,210 of 12,773 instances (17.3%) in the total model.

We now consider the confusion matrix with respect to walking, which had the lowest accuracy, and stroller, which had the highest accuracy. For the confusion matrix for walking (Table 12 middle-left), there was almost no output for all transportation modes except slopes and steps and most cases were output as a smooth surface. This confirms that detection of barrier details with walking using sensing is difficult. Qualitatively, there was a tendency to correctly detect slopes when descending with acceleration. For the confusion matrix of stroller (Table 12 middle-right), aside from mistaking a slope for a level most errors were between different slopes or among steps/projections of 8 cm or greater. There was a tendency to mistake an 8 cm step/projection as a 16 cm step/projection when large movements were used to cross it, but it should be possible to correctly detect these by collecting more data.

Examining changes in accuracy by adding noise, on the Smooth → Rough confusion matrix (Table 12, lower-left), level

Table 12 Confusion matrices for experiment 2 (barrier detection). No data for tactile paving on rough surfaces ware collected so values are zero. Hyphen indicates a barrier through which movement is impossible.

Smooth→Smooth, Simple

Actual label \ Estimated label	Level	2cm	8cm	16cm	5%	10%	Tactile	Stair	**Total**
Level	246,428	9	179	446	257	687	57	2,212	**250,275**
2cm	4,049	62	141	30	142	183	188	58	**4,853**
8cm	4,908	13	1,657	472	69	264	18	94	**7,495**
16cm	3,154	0	333	2,801	1	60	1	522	**6,872**
5%	18,839	8	55	41	1,672	1,351	40	359	**22,365**
10%	18,036	1	81	34	838	3,125	11	219	**22,345**
Tactile	3,658	12	8	0	14	9	688	66	**4,455**
Stair	5,653	0	0	14	0	25	0	9,567	**15,259**
Total	**304,725**	**105**	**2,454**	**3,838**	**2,993**	**5,704**	**1,003**	**13,097**	**333,919**

Smooth→Smooth, Total

Actual label \ Estimated label	Level	2cm	8cm	16cm	5%	10%	Tactile	Stair	**Total**
Level	251,795	91	570	841	1,526	2,679	197	3,930	**261,629**
2cm	3,392	265	300	47	472	382	235	97	**5,190**
8cm	2,840	94	3,209	696	73	415	23	133	**7,483**
16cm	1,772	6	403	3,991	59	91	7	547	**6,876**
5%	15,248	58	65	59	4,313	2,210	75	596	**22,624**
10%	14,728	26	164	12	1,934	6,434	30	695	**24,023**
Tactile	3,274	77	22	16	146	172	1,075	123	**4,905**
Stair	4,013	0	0	93	19	390	0	10,740	**15,255**
Total	**297,062**	**617**	**4,733**	**5,755**	**8,542**	**12,773**	**1,642**	**16,861**	**347,985**

Smooth→Smooth, Walking

Actual label \ Estimated label	Level	2cm	8cm	16cm	5%	10%	Tactile	Stair	**Total**
Level	30,304	0	59	141	357	1,383	0	2,535	**34,779**
2cm	803	0	4	9	7	68	0	81	**972**
8cm	845	0	1	24	22	93	0	78	**1,063**
16cm	646	0	14	38	1	36	0	206	**941**
5%	3,789	0	1	16	109	370	0	429	**4,714**
10%	4,092	0	0	10	92	448	0	437	**5,079**
Tactile	1,098	0	1	13	18	115	0	114	**1,359**
Stair	3,720	0	0	77	19	389	0	2,657	**6,862**
Total	**45,297**	**0**	**80**	**328**	**625**	**2,902**	**0**	**6,537**	**55,769**

Smooth→Smooth, Stroller

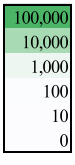
Actual label \ Estimated label	Level	2cm	8cm	16cm	5%	10%	Tactile	Stair	**Total**
Level	45,214	38	123	129	270	379	96	-	**46,249**
2cm	335	80	74	27	195	156	66	-	**933**
8cm	129	31	1,053	345	6	46	0	-	**1,610**
16cm	179	3	218	2,097	4	47	4	-	**2,552**
5%	1,829	46	40	1	1,160	913	19	-	**4,008**
10%	855	23	101	1	498	2,626	16	-	**4,120**
Tactile	118	10	4	0	28	2	563	-	**725**
Stair	-	-	-	-	-	-	-	-	**-**
Total	**48,659**	**231**	**1,613**	**2,600**	**2,161**	**4,169**	**764**	**-**	**60,197**

Smooth→Rough, Stroller

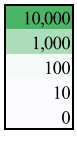
Actual label \ Estimated label	Level	2cm	8cm	16cm	5%	10%	Tactile	Stair	**Total**
Level	6,558	437	105	70	1,429	277	3,328	-	**12,204**
2cm	76	8	563	341	125	558	80	-	**1,751**
8cm	136	19	658	266	19	32	29	-	**1,159**
16cm	184	4	508	782	20	25	7	-	**1,530**
5%	569	70	346	128	1,475	1,592	763	-	**4,943**
10%	295	67	1,403	659	1,530	2,553	631	-	**7,138**
Tactile	0	0	0	0	0	0	0	-	**0**
Stair	-	-	-	-	-	-	-	-	**-**
Total	**7,818**	**605**	**3,583**	**2,246**	**4,598**	**5,037**	**4,838**	**-**	**28,725**

Noise added→Rough, Stroller

Actual label \ Estimated label	Level	2cm	8cm	16cm	5%	10%	Tactile	Stair	**Total**
Level	8,132	419	62	56	1,244	262	2,029	-	**12,204**
2cm	114	7	631	235	146	562	56	-	**1,751**
8cm	150	39	719	205	24	20	2	-	**1,159**
16cm	210	4	445	832	29	10	0	-	**1,530**
5%	819	32	261	48	1,691	1,621	471	-	**4,943**
10%	524	37	1,128	642	1,775	2,619	413	-	**7,138**
Tactile	0	0	0	0	0	0	0	-	**0**
Stair	-	-	-	-	-	-	-	-	**-**
Total	**9,949**	**538**	**3,246**	**2,018**	**4,909**	**5,094**	**2,971**	**-**	**28,725**



Simple model / Total model



Walking / Stroller

© 2020 Information Processing Society of Japan

Table 13 Experiment 3 results (barrier detection task on rough surface when the number of participants was changed from half to one-ninth). Higher values than proposed method are in bold.

Model	Proposed	Conventional								
		1/9	1/8	1/7	1/6	1/5	1/4	1/3	1/2	1/1
Walking	0.314	0.225	0.205	0.242	0.248	0.268	0.268	**0.362**	**0.365**	**0.367**
Wheelchair	0.576	**0.583**	**0.722**	**0.745**	**0.703**	**0.765**	**0.740**	**0.724**	**0.754**	**0.783**
Stroller	0.516	0.323	0.344	0.399	0.399	0.411	0.471	**0.517**	**0.577**	**0.636**
Handcart	0.358	0.129	0.102	0.150	0.142	0.247	0.247	0.285	**0.376**	**0.467**
Suitcase	0.400	**0.396**	**0.406**	**0.466**	**0.454**	**0.455**	**0.442**	**0.448**	**0.510**	**0.573**
Bicycle	0.505	**0.516**	**0.577**	**0.658**	**0.668**	**0.620**	**0.654**	**0.692**	**0.729**	**0.766**

was often misdetected as tactile paving, and overall there were more errors. However, for Noise-added → Rough (Table 12, middle-right), level cases misdetected as tactile paving decreased significantly, the proportion of cases outputting level increased, and the overall number of errors for barriers decreased.

5.3 Experiment 3: VS Conventional Augmentation Method

Table 13 shows the accuracy of barrier detection on rough surfaces when the number of participants was changed from half to one-ninth.

When the results were checked for the stroller and handcart with high accuracy in Experiment 2, the results were almost the same when the number of participants using stroller and handcart were reduced to 1/3 and 1/2, respectively. This means that the data-collection cost of the proposed method is lower when there are three or more types of road surfaces when using a stroller and there are two or more types of road surfaces when using a handcart. The proposed method therefore proves effective for handling many road surfaces. In addition, when collecting data on various road surfaces even with a small number of participants, various road routes must be prepared and much time is required so the proposed method offers the advantage of not requiring preparing routes.

6. Conclusion and Future Prospects

We propose a model that detects transportation modes and types of barriers in two steps, to detect details about barriers for various transportation modes using sensor data. We also propose a method for reducing the cost of collecting data by collecting data on a course with smooth surfaces and simulating rough surfaces by adding noise to these data. In experiments on detecting the six transportation modes, our method achieved 91.5% accuracy, and the addition of noise increased accuracy by 3.7 percentage points for rough surfaces. When detecting the type of barrier from eight classes, the method achieved an accuracy of 87.7% when using a stroller, and addition of noise increased accuracy by 6.8 percentage points for rough surfaces. We show the effectiveness of the proposed method and detected barrier details using multiple transportation modes.

In actual use it is necessary to consider various conditions that were not considered in this paper. For example, when the moving speed significantly differs such as running, when there is extremely heavy luggage, when the vibration characteristics significantly differ such as using a simple stroller, and when the performance of a sensor is low such as when using a budget smartphone. We used a simple method of data augmentation by adding noise, but it is also conceivable to generate data under various conditions by using a conditional generation model such as generative adversarial networks (GAN) [27]. We will work to improve detection accuracy by matching GPS to walking paths and we will study ways to integrate data from multiple persons and multiple transportation modes. We will also conduct experiments using real sensor data collected in a town, including various road surfaces such as asphalt, brick, gravel, and mud.

References

[1] Kurauchi, Y., Abe, N., Konishi, H. and Seshimo, H.: Barrier Detection using Sensor Data from Multiple Transportation modes with Data Augmentation, *IEEE COMPSAC*, pp.667–675 (2019).

[2] Arai, K., Miura, C. and Kobayashi, T.: Slope Information Collection System Using Sensor Information from General-Purpose Wheelchair Users, *IEEE ICCE*, pp.1–2 (2019).

[3] Miyata, A., Araki, I. and Wang, T.: Barrier detection using sensor data from unimpaired pedestrians, *HCII*, Vol.10908, pp.308–319 (2018).

[4] Ordonez, F.J. and Roggen, D.: Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition, *Sensors*, Vol.16, No.1, pp.115 (2016).

[5] Yang, J.B., Nguyen, M.N., San, P.P., Li, X. and Krishnaswamy, S.: Deep convolutional neural networks on multichannel time series for human activity recognition, *IJICAI*, pp.3995–4001 (2015).

[6] Gravina, R., Alinia, P., Ghasemzadeh, H. and Fortino, G.: Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges, *Information Fusion*, Vol.35, pp.68–80 (2017).

[7] Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S. and Balakrishnan, H.: The Pothole Patrol: Using a mobile sensor network for road surface monitoring, *ACM MobiSys*, pp.29–39 (2008).

[8] Elhamshary, M., Youssef, M., Uchiyama, A. and Yamaguchi, H.: CrowdMeter: Congestion level estimation in railway stations using smartphones, *IEEE PerCom*, pp.1–12 (2018).

[9] Wang, Q., Guo, B., Liu, Y. and Han, Q.: CrowdNavi: Last-mile outdoor navigation for pedestrians using mobile crowdsensing, *ACM HCI*, Vol.2, No.179 (2018).

[10] Yamamoto, C., Funakoshi, K., Konishi, H., Ochiai, K. and Kawanobe, A.: Technology for social development of accessibility maps, *NTT Technical Review*, Vol.14, No.7 (2016).

[11] Miura, T., Yabu, K., Ikematsu, S. and Kano, A.: Barrier-free walk: A social sharing platform of barrier-free information for sensory/physically-impaired and aged people, *IEEE SMC*, pp.2927–2932 (2012).

[12] Saha, M., Saugstad, M., Maddali, H.T., Zeng, A., Holland, R., Bower, S., Dash, A., Chen, S., Li, A., Hara, K. and Froehlich, J.: Project Sidewalk: A Web-based Crowdsourcing Tool for Collecting Sidewalk Accessibility Data at Scale, *ACM CHI*, No.62 (2019).

[13] Yamato, Y., Go, K. and Miyata, A.: Gamification Approach for Gathering Barrier Information, *CollabTech 2019*, pp.35–38 (2019).

[14] Jain, S., Borgiattino, C., Ren, Y., Gruteser, M., Chen, Y. and Chiasserini, C.-F.: Lookup: Enabling pedestrian safety services via shoe sensing, *ACM Mobisys*, pp.257–271 (2015).

[15] Taniue, H., Kaneko, J., Kojima, K., Fulwani, D. and Kumar, S.: Development of automatic barrier detection system for wheelchair, *IEEE GCCE*, pp.374–376 (2015).

[16] Cha, Y.J., Choi, W. and Büyüköztürk, O.: Deep learning-based crack damage detection using convolutional neural networks, *Computer-Aided Civil and Infrastructure Engineering*, Vol.32, No.5, pp.361–378 (2017).

© 2020 Information Processing Society of Japan

[17] Endo, Y., Toda, H., Nishida, K. and Kawanobe, A.: Deep feature extraction from trajectories for transportation mode estimation, *PAKDD*, Vol.9652 (2016).
[18] Lorintiu, O. and Vassilev, A.: Transportation mode recognition based on smartphone embedded sensors for carbon footprint estimation, *IEEE ITSC*, pp.1976–1981 (2016).
[19] DeVries, T. and Taylor, G.W.: Improved regularization of convolutional neural networks with cutout, arXiv preprint arXiv:1708.04552 (2017).
[20] Edunov, S., Ott, M., Auli, M. and Grangier, D.: Understanding back-translation at scale, arXiv preprint arXiv:1808.09381 (2018).
[21] Um, T.T., Pfister, F.M.J., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U. and Kulić, D.: Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks, *ACM ICMI*, pp.216–220 (2017).
[22] Zisselman, E., Adler, A. and Elad, M.: Compressed Learning for Image Classification: A Deep Neural Network Approach, *Processing, Analyzing and Learning of Images, Shapes, and Forms*, Vol.19 (2018).
[23] Kang, Y., Hauswald, J., Gao, C. and Rovinski, A.: Neurosurgeon: Collaborative intelligence between the cloud and mobile edge, *CM SIGPLAN Notices*, Vol.52, No.4, pp.615–629 (2017).
[24] Ministry of Land, Infrastructure, and Transport and Tourism of Japan: Development specification for spatial network model for pedestrians, available from ⟨http://www.mlit.go.jp/sogoseisaku/soukou/seisakutokatsu_soukou_tk_000026.html⟩ (accessed 2019-10-25).
[25] Aubanel, V., Lecumberri, M.L.G. and Cooke, M.: The Sharvard Corpus: A phonemically-balanced Spanish sentence resource for audiology, *International Journal of Audiology*, Vol.53, No.9, pp.1–6 (2014).
[26] GitHub, available from ⟨https://github.com/hyperopt/hyperopt⟩ (accessed 2019-10-25).
[27] Goodfellow, I.J., Bengio, Y., et al.: Generative Adversarial Nets, arXiv preprint arXiv:1406.2661 (2014).

Yuki Kurauchi received his B.E., M.S.E. degrees in Keio University, Japan, in 2008 and 2010. He joined NTT Corp. in 2010. He worked at NTT DOCOMO, INC. from 2013 to 2017. He had been engaged in research on music informatics/twitter mining and planning of marketing strategy for mobile phone business. His current research interests include computational sensing and machine learning.

Naoto Abe received his Ph.D. degree in Hokkaido University, Japan, in 2006. He joined NTT Corp. in 2006. He was engaged in web mining research, cloud storage service planning/development/operation, and privacy impact assessment activities. He is currently engaged in research and development of accessibility information collection technology using crowd sensing. He is a member of IEICE, IPSJ.

Osamu Matsuda received his B.S. degree in Aoyama Gakuin University, Japan, in 1994 and joined NTT Corp. the same year. He was engaged in service development of intelligent transport system, interactive visual communication, geographic information system, and satellite imagery analysis. His current research interests include sensing and modeling the surrounding space and environment using geospatial information technology.

Hitoshi Seshimo received a B.E. and M.E. in mechanical engineering from Waseda University, Tokyo, in 1995 and 1997. He joined NTT in 1997. His research interests include computer aided instruction, web-based learning, content distribution and navigation systems, and geographical information services.

© 2020 Information Processing Society of Japan