PAPER

Bicycle Behavior Recognition Using 3-Axis Acceleration Sensor and 3-Axis Gyro Sensor Equipped with Smartphone

Yuri USAMI[†a)], **Kazuaki ISHIKAWA**[††], **Toshinori TAKAYAMA**[††], ***Nonmembers***, **Masao YANAGISAWA**[†], ***Fellow***, ***and*** **Nozomu TOGAWA**[†b)], ***Senior Member***

SUMMARY It becomes possible to prevent accidents beforehand by predicting dangerous riding behavior based on recognition of bicycle behaviors. In this paper, we propose a bicycle behavior recognition method using a three-axis acceleration sensor and three-axis gyro sensor equipped with a smartphone when it is installed on a bicycle handlebar. We focus on the periodic handlebar motions for balancing while running a bicycle and reduce the sensor noises caused by them. After that, we use machine learning for recognizing the bicycle behaviors, effectively utilizing the motion features in bicycle behavior recognition. The experimental results demonstrate that the proposed method accurately recognizes the four bicycle behaviors of stop, run straight, turn right, and turn left and its F-measure becomes around 0.9. The results indicate that, even if the smartphone is installed on the noisy bicycle handlebar, our proposed method can recognize the bicycle behaviors with almost the same accuracy as the one when a smartphone is installed on a rear axle of a bicycle on which the handlebar motion noises can be much reduced.

key words: *behavior recognition, smartphone, acceleration sensor, gyro sensor, bicycle*

1. Introduction

1.1 Bicycle Behavior Recognition

It becomes possible to prevent accidents beforehand by predicting dangerous riding behavior based on recognition of automobile/motorcycle behaviors [1]–[3]. On the other hand, according to the bicycle accident analysis data from the National Police Agency in Japan [4], the number of bicycle accidents in Tokyo area, Japan, became 11,901 in 2017, which occupies around one third of all the traffic accidents in Tokyo. The National Police Agency sets up the rules for bicycle riding and aims to reduce dangerous riding but not so many users comply with the rules [5]. In order to reduce the bicycle accidents, it is quite necessary to recognize bicycle behaviors and prevent dangerous riding beforehand [6]–[8] but few researches have been reported so far to recognize bicycle behaviors[*].

While we always use roadways when riding motorcycles, we can use roadways and sometimes sidewalks as well in Japan when riding bicycles. Furthermore, since many bicycle users tend not to comply with the traffic rules and are likely to run outside the roadways, there are many motions to avoid obstacles such as pedestrians and utility poles. When we apply the behavior recognition method for motorcycles [1]–[3] to bicycles, it must be hard to recognize accurately these bicycle avoiding behaviors. Furthermore, since bicycles are balanced by keeping their handlebar steadily turning left or right while running, we cannot ignore the noises caused by their handlebar motions [9], [10].

1.2 Classification of Existing Works

The bicycle behavior recognition methods proposed so far can be roughly classified into those using dedicated sensor devices and those using a smartphone-mounted sensors:

1. **Bicycle behavior recognition methods using dedicated sensor devices [6], [11]:**
   In these methods, the dedicated sensor devices are attached to a bicycle in advance and data obtained from them are used to recognize the bicycle behaviors. Smaldone et al. propose a system which detects approaching vehicles by using audio/video sensors [6]. Yamanaka et al. propose a method which recognizes bicycle behavior by using many sensing devices such as an video camera, a data logger, a lateral distance sensor, a steering sensor, a braking sensor, a vibration sensor and a speed sensor on a bicycle [11]. All of these methods require the dedicated sensor devices on a bicycle for recognizing the bicycle behaviors and thus they require many costs to recognize bicycle behaviors.
2. **Bicycle behavior recognition methods using a smartphone-mounted sensors [8]:**
   In [8], a method to recognize bicycle behaviors is proposed by using the data from the three-axis acceleration sensor equipped with a smartphone. In this method, we firstly install the smartphone beside the rear wheel axle of the bicycle. Then the influence of noises caused by the handlebar motions can be minimized and thus we do not have to reduce the sensor noises furthermore. We can directly obtain motion feature values from sensor data at the time of right/left turn. However, users cannot use smartphones even while stopping.

In this paper, we focus on installing a smartphone on a

Manuscript received July 23, 2018.
Manuscript revised January 11, 2019.

[†]The authos are with Dept. of Computer Science and Communications Engineering, Waseda University, Tokyo, 169-8555 Japan.
[††]The authos are with Zenrin DataCom Co., LTD., Tokyo, 108-6206 Japan.
a) E-mail: yuri.usami@togawa.cs.waseda.ac.jp
b) E-mail: togawa@togawa.cs.waseda.ac.jp

DOI: 10.1587/transfun.E102.A.953

[*]How to predict dangerous bicycle behavios using four bicycle behaviors (stop, run straight, turn right, and turn left) will be discussed in Sect. 5.1.

bicycle handlebar and aim to recognize its behavior without using any other sensor devices. When installing a smartphone on a bicycle handlebar, it makes the sensor values more susceptible to noise and recognizing bicycle behaviors themselves will be more difficult. The largest problem here is how to recognize bicycle behaviors accurately by reducing the influence of the handlebar motions.

1.3 Our Proposal

Based on the discussion above, we propose a bicycle behavior recognition method using three-axis acceleration sensor and three-axis gyro sensor equipped with a smartphone[†]. We focus on the periodic handlebar motions for balancing while running a bicycle and reduce the sensor noises caused by them. After that, we learn the bicycle behaviors using machine learning by effectively extracting the bicycle motion features, where we use a random forest classifier. Finally, we recognize bicycle behaviors based on the learned random forest classifier. We have evaluated our proposed bicycle behavior recognition and confirms its efficiency and effectiveness.

1.4 Contributions of the Paper

The contributions of this paper are summarized as follows:

1. We focus on the periodic handlebar motions for balancing while running a bicycle and the sensor noises caused by them are effectively removed. Furthermore, we learn the bicycle behaviors using machine learning by extracting the five-dimensional bicycle motion features and hence we can accurately recognize the bicycle behaviors.
2. We have applied our proposed method to training data composed of 8,969 data and test data composed of 295 data. Then the four bicycle behavior of stop, run straight, turning right, and turning left are accurately recognized. The F-measure becomes high enough which becomes around 0.9 in all the four behaviors. The results indicate that, even if the smartphone is installed on the noisy bicycle handlebar, our proposed method can recognize the bicycle behaviors with almost the same accuracy as the one when a smartphone is installed on a rear axle of a bicycle on which the handlebar motion noises can be much reduced.

1.5 Organization of this Paper

The rest of this paper is organized as follows: Section 2 defines a bicycle behavior recognition problem; Section 4 proposes our bicycle behavior recognition method using smartphone sensors where we first propose noise reduction processes based on filtering and then we propose a bicycle behavior recognition method; Section 4 demonstrates evaluation experiment. Section 5 discusses an application scenario of the proposed method and several supplementary experiments and Sect. 6 gives concluding remarks.

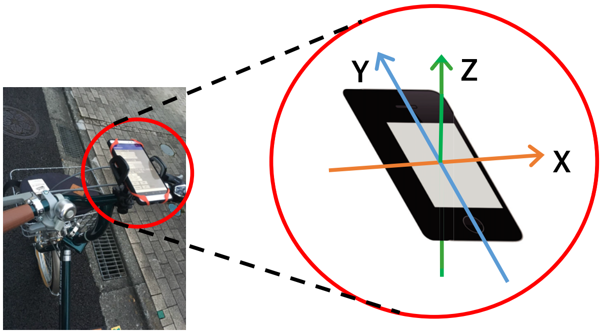


Fig. 1 The installation position of a smartphone and the axies of the acceleration/gyro sensors equipped with a smartphone.

2. Bicycle Behavior Recognition Problem

In this section, we define a bicycle behavior recognition problem. We assume that a smartphone with a three-axis acceleration sensor and a three-axis gyro sensor is used. The installation position of a smartphone and the axies of the acceleration sensor and gyro sensor equipped with a smartphone are shown in Fig. 1. The three-axis acceleration sensor can obtain the X-axis, Y-axis, and Z-axis acceleration values. The smartphone is installed on the bicycle with the Y-axis directing forward and the screen upward as shown in Fig. 1. Figure 1 shows that the installation position of a smartphone is the bicycle handlebar and the smartphone is located horizontally. Then the acceleration sensor obtains a Y-axis positive value when the bicycle is accelerated. It obtains a negative value when the bicycle is decelerated. The acceleration sensor obtains an X-axis positive value (or negative value) when a bicycle turns to the right (or the left). The acceleration sensor obtains a Z-axis positive value (or negative value) when a bicycle moves upward (or downward). The gyro sensor obtains a positive value when it turns counterclockwise around the axis and it becomes a negative value when it turns clockwise.

The user installs his/her own smartphone on the bicycle handlebar as shown in Fig. 1 and rides the bicycle. Let a_x, a_y and a_z be the X-axis, Y-axis, and Z-axis values obtained from the acceleration sensor, respectively. Let ω_x, ω_y and ω_z be the X-axis, Y-axis, and Z-axis values obtained from the gyro sensor, respectively. These values are obtained every t_s seconds (in this paper, we assume $t_s = 0.05$s). We set the time window size to T seconds and recognize the bicycle behaviors of stop, run straight, turn right, and turn left, from the values of the three-axis acceleration sensor and three-axis gyro sensor for every time window (see Fig. 2)[††]. Then, the bicycle behavior recognition problem is defined as follows:

[†] The preliminary version of this paper appeared in [12].

[††] Actually, as discussed in Sect. 3.2.2, we apply a sliding window scheme to our method in which every time window overlaps each other.

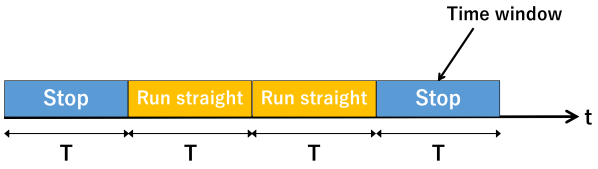


Fig. 2 Time windows and behavior recognition.

Definition 1 (Bicycle behavior recognition problem)**.** *When the values of three-axis acceleration sensor and three-axis gyro sensor are given every t_s seconds from the smartphone sensors installed on the bicycle handlebar, the bicycle behavior recognition problem is to recognize a bicycle behavior (stop, run straight, turn right, or turn left) at every time window of T seconds.*

3. Bicycle Behavior Recognition Using Smartphone Sensors

When recognizing bicycle behaviors with smartphone sensors on the bicycle handlebar, the values from the three-axis acceleration sensor and those from the three-axis gyro sensor always include the following noises [9], [13]:

(A) Noises from road surface (both of acceleration sensor and gyro sensor)
(B) Noises due to handlebar motions (both of acceleration sensor and gyro sensor)
(C) Noises due to drifts (gyro sensor)

(A) The noises from road surface can be reduced by applying a low pass filter (LPF) to obtained data because the frequency of the noises are higher than that of bicycle behaviors to be recognized. We can also reduce these noises by limiting the axis of the sensor to be used. The detailed discussions are given just below.

(B) The noises due to handlebar motions strongly affect the acceleration sensor and the gyro sensor, which are mainly caused by pedaling.

(C) The noise due to drifts strongly affect the gyro sensor. In order to eliminate the drift noises of the gyro sensor, we usually calculate the angle data from the acceleration sensor which are not affected by drift noises. After that, we compare the values with the ones obtained from the gyro sensor and finally know how much drift noises are included in the gyro sensor [14].

For example, let us assume that the acceleration sensor and gyro sensor are located horizontally. In this case, the gravitational acceleration works only in Z-axis. The acceleration works in Y-axis when the acceleration sensor rotates around X-axis, and the acceleration works in X-axis when the acceleration sensor rotates around Y-axis. By using these acceleration values, we can calculate the angles of the X-axis and the Y-axis utilizing only the acceleration sensor, and the drift noises of the gyro sensor can be removed based on these values.

However, when the acceleration sensor rotates around Z-axis, the gravitational acceleration continues to work only in Z-axis. Therefore, we cannot calculate the angles of Z-axis using only the acceleration sensor, and the drift noises of the gyro sensor cannot be reduced in this case. As described later, the proposed method requires the Z-axis values of the gyro sensor and hence the correction process above cannot be applied there.

According to the conventional bicycle behavior recognition method using smartphone sensors, the noises of (A) and (B) can be minimized when the smartphone is installed at the side of the rear wheel axle of the bicycle [15]. Thus, in [8], bicycle behaviors are recognized by using the smartphone acceleration sensor installed there.

On the other hand, if a smartphone is installed on the bicycle handlebar, the noises of (B) above strongly affect both the acceleration sensor and the gyro sensor since the bicycle is balanced by keeping its handlebar steadily turning left or right while running. It must be more difficult to accurately recognize the bicycle behaviors in this case.

Based on the discussions above, we propose a method of (1) reducing the influence of noises of (A), (B) and (C) by performing noise reduction processes based on filtering and (2) recognizing bicycle behaviors based on machine learning.

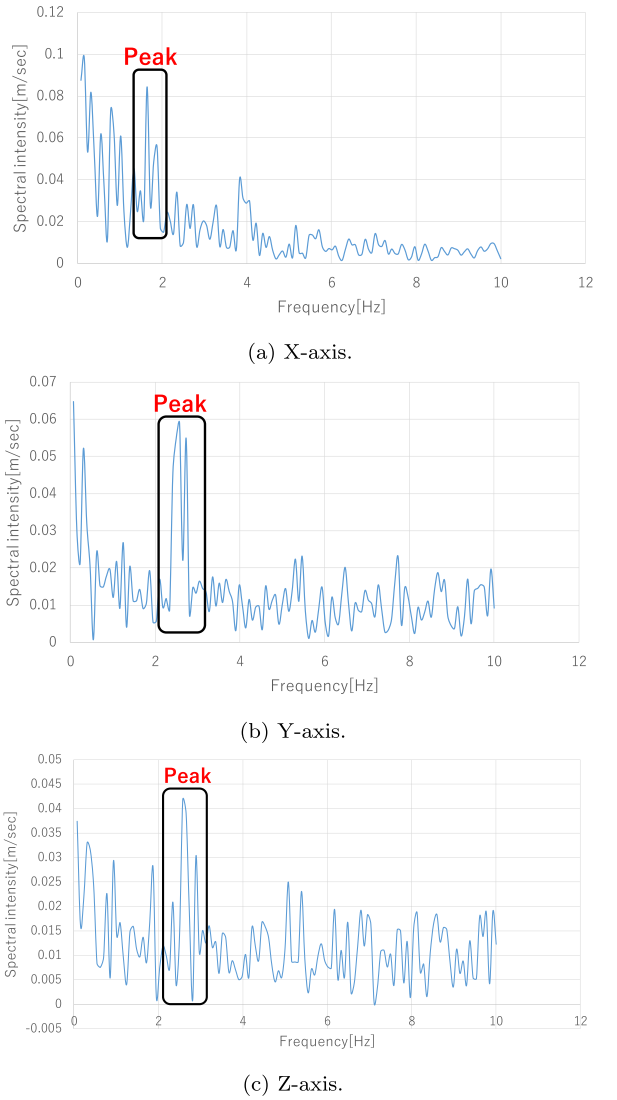
3.1 Noise Reduction Processes Based on Filtering

Now we propose noise reduction processes based on filtering for reducing the noises of (A), (B) and (C).

3.1.1 Filtering for Three-Axis Acceleration Sensor

The noises of (A) and (B) affect the three-axis acceleration sensor. In order to investigate how much these noises affect the three-axis acceleration sensor values, we conducted a preliminary experiment. We installed the smartphone on a bicycle handlebar and ran the bicycle straight for approximately 50m on a flat road. After that, we ran the bicycle randomly meandering on the same road. The smartphone used was HUAWEI Mate 9 [16] (we use the same smartphone in the experiments later). Note that we remove the gravitational acceleration value (vertically downward, 9.8 m/s^2) from the three-axis acceleration values beforehand throughout the paper.

Figure 3 shows the frequency distribution of the three-axis acceleration sensor values while running straight. Figure 4 shows the frequency distribution while meandering. From these figures, we can observe the peak values of around 1.5 Hz to 2.0 Hz while running straight, which shows the noises from (A) and (B). We can also observe the peak values of around 0.5 Hz while meandering, which shows left-turns or right-turns. The frequencies caused by the noises of (A) and (B) are higher than those caused by turning left or right. Hence, by appropriately applying LPF to the three-axis acceleration sensor values, we can remove the noises of (A) and (B) simultaneously. Then we set the cut-off frequency of LPF to 1Hz and apply it to the acceleration sensor values. In LPF, we use the Hamming window as a window function.

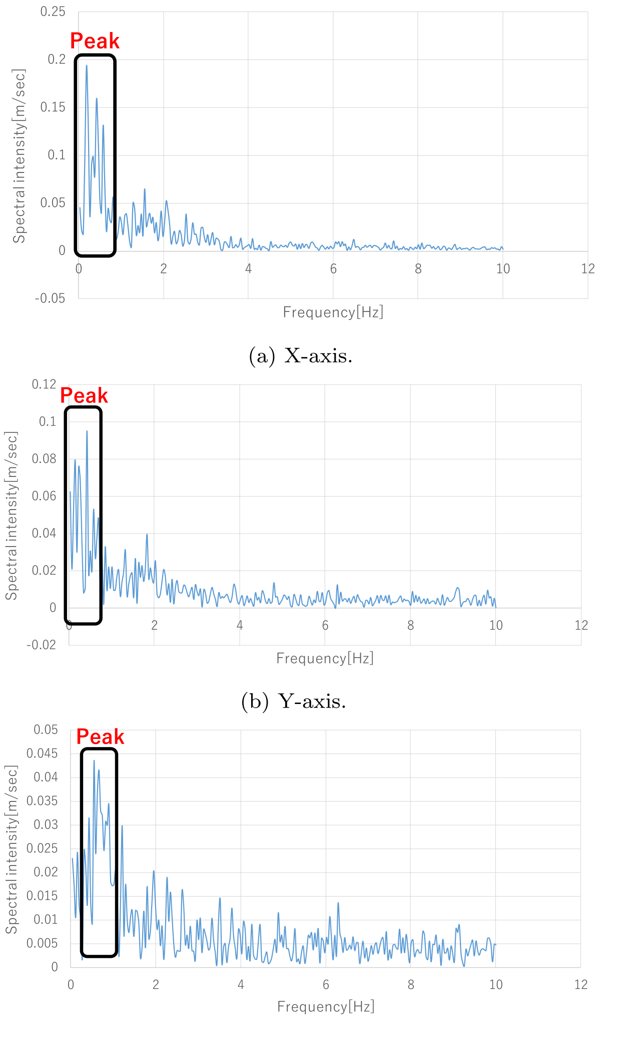


(a) X-axis.

(b) Y-axis.

(c) Z-axis.

Fig. 3 Frequency distribution of acceleration sensor values while straight running.



(a) X-axis.

(b) Y-axis.

(c) Z-axis.

Fig. 4 Frequency distribution of acceleration sensor value while meandering.

The number of samples in every Hamming window is set to be 19 based on [1].

In order to investigate how much the LPF reduces the noises, we also conducted another preliminary experiment. We ran a bicycle counterclockwise on the experimental route shown in Fig. 5[†] and obtained the acceleration sensor values. Now we pick up the X-axis values as an example. We set the size of the time window T_i $(i = 0, 1, 2, \cdots)$ to 1.6 seconds. Since the sensor obtains the acceleration values every t_s = 0.05 seconds, every T_i contains 32 acceleration values in X-axis. When we apply the LPF to these 32 values, we also have 32 low-pass filtered values. Since we use the sliding window method, every time window T_i overlaps half to the next time window T_{i+1} (See Sect. 3.2.2). By averaging the 32 values in each time window, we can obtain the average X-axis value in every time window. We can also obtain the average Z-axis value in every time window[††].

Figure 6 shows the average X-axis acceleration value per time window. Figure 7 shows the average Z-axis acceleration value per time window.

In Fig. 6, the bicycle turns to the left at the time window numbers of 23–30, 52–58, 97–103 and 122–129. The amplitude of the X-axis acceleration value after noise reduction (Fig. 6(b)) tends to increase in a very short period of time at the time of left turn. Similarly, the bicycle turns to the left at the time window numbers of 23–30, 52–58, 97–103 and 122–129 in Fig. 7. The amplitude of the Z-axis acceleration value after noise reduction (Fig. 7(b)) tends to increase in a very short period of time at the time of left turn.

As can be seen from the figures, we expect that the ac-

[†] As described in Sect 4, the experimental route is a residential district in Tokyo, Japan, with sidewalks and telegraph poles, partly sloping. How the slopes in the experimental route affects the experiments will be discussed in Sect. 5.3.

[††] As discussed in Sect. 3.2.1, we finally use the average X-axis values and the average Z-axis values from the acceleration sensor as the feature values and do not use average Y-axis values.

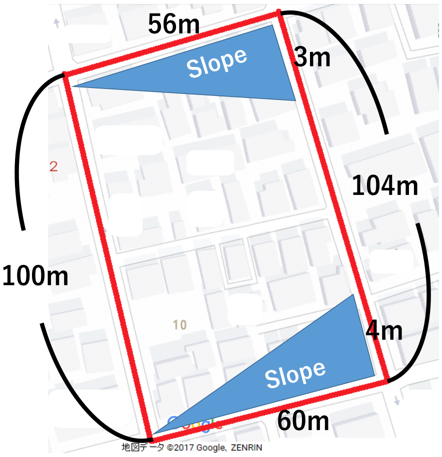
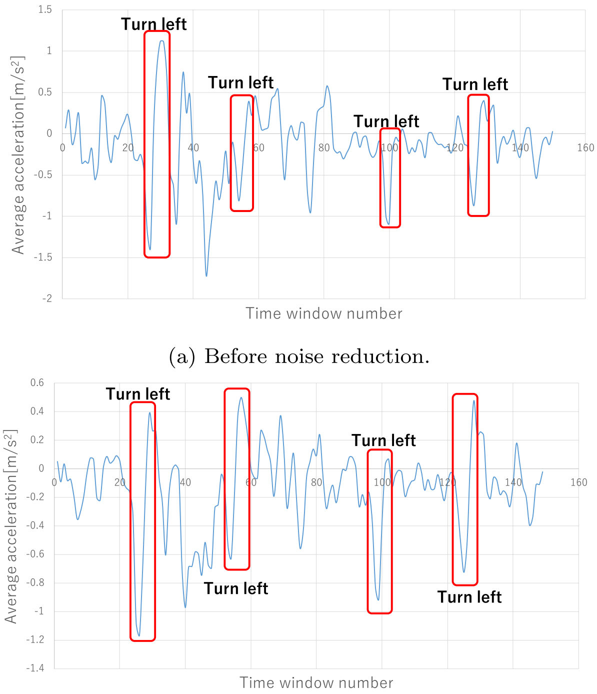


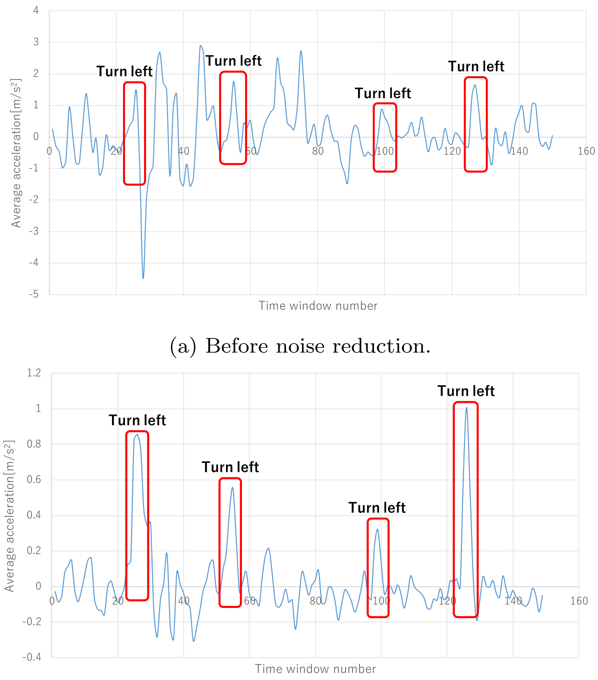
Fig. 5 Experimental route.



(a) Before noise reduction.

(b) After noise reduction.

Fig. 6 Average X-axis acceleration value per time window (time window size = 32).



(a) Before noise reduction.

(b) After noise reduction.

Fig. 7 Average Z-axis acceleration value per time window (time window size = 32).

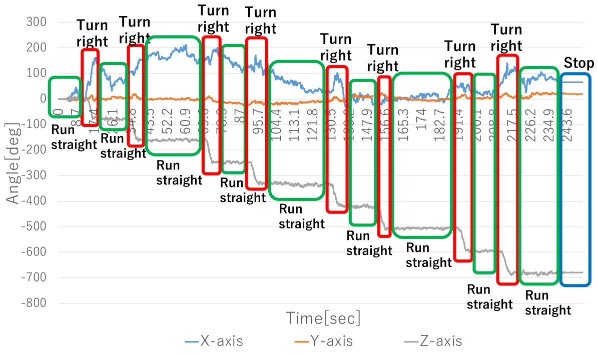


Fig. 8 Angle difference values obtained from each axis of the gyro sensor.

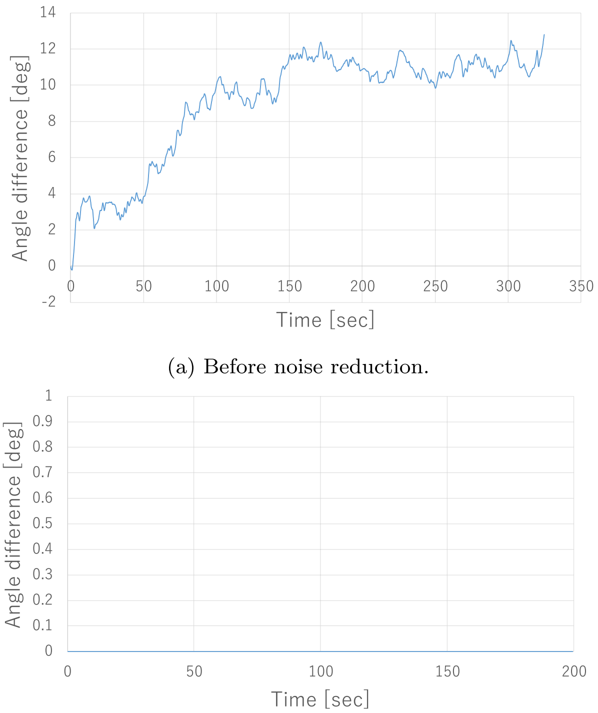
curacy of the bicycle behavior recognition becomes high by using the acceleration value after noise reduction processes.

We can also observe the similar features in other bicycle behaviors such as stop and turning right when we use low-pass-filtered acceleration values.

3.1.2 Filtering for Three-Axis Gyro Sensor

All the noises of (A), (B), and (C) affect the three-axis gyro sensor. When the smartphone is located horizontally and we focus only on its Z-axis gyro sensor values, the noises of (A) cannot affect them since they are always up-and-down motions on a rough road surface. Moreover, the X-axis and Y-axis values of the gyro sensor do not contribute to distinguish between running straight and turning right/left as shown in Fig. 8. We cannot well see the differences in the X-axis and Y-axis angles when the bicycle stops, runs straight, and turns to the right. But we can see the clear differences in the Z-axis angles here. Hence we use the Z-axis gyro sensor values to recognize bicycle behaviors.

Now we conducted a preliminary experiment to investigate how much Z-axis values of the gyro sensor are affected by (C) drift noises when locating the smartphone on a horizontal desk with its screen upward. The gyro sensor obtained the Z-axis angular velocity $\omega_z(i)$ $(i = 0, 1, 2, \cdots)$ every $t_s = 0.05$ seconds and we can calculate the angle difference $d_z(i)$ as below [17]:



(a) Before noise reduction.

(b) After noise reduction.

Fig. 9 Drift of Z-axis in the gyro sensor.

$$\begin{cases} d_z(0) = 0 \\ d_z(i) = \dfrac{1}{2} t_s \times (\omega_z(i-1) + \omega_z(i)) \quad (i = 1, 2, \cdots) \end{cases} \tag{1}$$

Figure 9(a) shows the angle values before noise reduction. When we just located the smartphone on a horizontal desk with the screen upward, the angle was gradually increased. At that time, the maximum angle difference value during 0.05 seconds became 0.09 degrees. Hence, if we set up the threshold value to be 0.09, we can remove the drift noise as in Eq. (2) below:

$$d'_z(i) = \begin{cases} 0 & (d_z(i) \leq 0.09) \\ d_z(i) & (d_z(i) > 0.09) \end{cases} \tag{2}$$

Next, in order to investigate how much the noises of (B) affect Z-axis values of the gyro sensor, we ran a bicycle straight on a flat road and observe the angle difference $d_z(i)$ every 0.05 seconds.

Since the bicycle is balanced by keeping its handlebar periodically turning left or right, we can set up the threshold value d_{th} so that, if $d_z(i)$ becomes larger than d_{th}, we can consider it that the bicycle actually changes its direction. However, if $d_z(i)$ does not become larger than d_{th}, it means that the bicycle does not actually change its direction but it just meanders. In this case, the angle difference $d_z(i)$ can be calculated by:

$$d'_z(i) = \begin{cases} 0 & (d_z(i) \leq d_{th}) \\ d_z(i) & (d_z(i) > d_{th}) \end{cases} \tag{3}$$

Since the average angle difference value during 0.05 seconds in the preliminary experiment was 0.49 degrees, we set the threshold value d_{th} to be $d_{th} = 0.49$.

To summarize the above, since Eq. (3) includes Eq. (2), we can reduce the noises of (B) and the noises of (C) simultaneously, just using Eq. (3). When we apply Eq. (3) to Z-axis values of the gyro sensor, we can obtain the flat wave form as depicted in Fig. 9(b), where all the noises of (A), (B), and (C) are reduced.

Note that, as above, the noises due to handlebar motions are too much larger than the drift noises. Even if the drift noises differ depending on the individual difference in gyro sensors, they must be still too small compared to the noises due to handlebar motions.

Hence, the noise reduction processes based on filtering for gyro sensors given by Eq. (3) must be effective regardless of the individual difference in gyro sensors.

3.2 Bicycle Behavior Recognition Method

Since the bicycle behaviors depend on riding situation, it must be very hard to set fixed threshold values for each bicycle behavior. Then we utilize a machine learning approach which can analyze many bicycle running data and recognize the current behavior based on them.

There have been proposed many machine learning approaches such as a k-nearest neighborhood method [18] and a support vector machine method [19]. However, bicycle behaviors should be correctly recognized by the values from the acceleration sensor and gyro sensor, even if some of them include the outlier values. Hence, we select the random forest classifier [20], which is based on ensemble learning and utilizes a group of weak learners.

When recognizing bicycle behaviors by machine learning, we have to design bicycle behavior features which can effectively classify the bicycle behaviors.

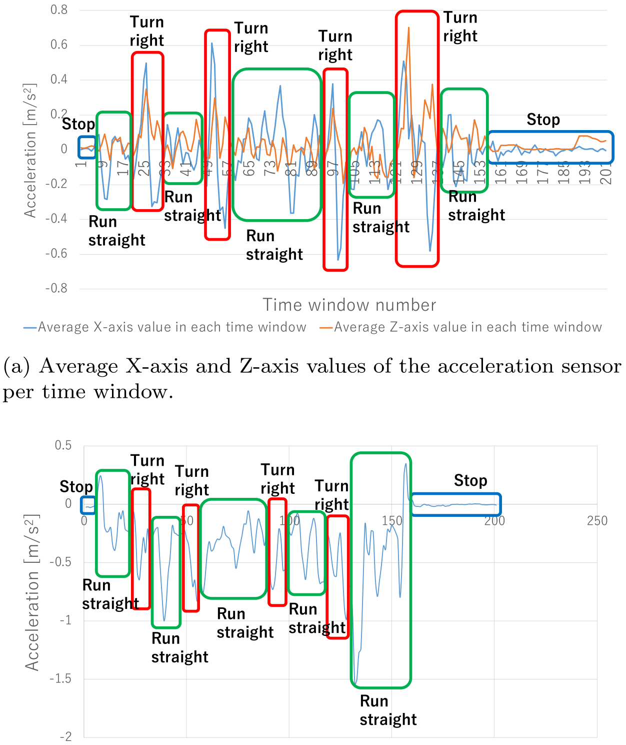
3.2.1 Bicycle Behavior Features

Through several preliminary experiments, we observe which ones of the acceleration sensor values and gyro sensor values most contribute to classify the bicycle behaviors.

(1) Acceleration Sensor Values

We installed a smartphone on the bicycle handlebar with its Y-axis directing forward (see Fig. 1) and ran the bicycle around the experimental route shown in Fig. 5 clockwise. We obtained the X-axis, Y-axis, and Z-axis values of the acceleration sensor every $t_s = 0.05$ seconds. For every time window whose size is 1.6 seconds, the obtained values are averaged and we have averaged X-axis, Y-axis, and Z-axis values, which are plotted in Figs. 10(a) and 10(b). Note that we use here noise-reduced values and every time window overlaps half to the next time window.

The average X-axis and Z-axis values of the acceleration sensor becomes closer to zero when the bicycle stops and the values fluctuate periodically around zero when running.



(a) Average X-axis and Z-axis values of the acceleration sensor per time window.

(b) Average Y-axis value of the acceleration sensor per time window.

Fig. 10 How much the acceleration sensor values contribute to bicycle behaviors.

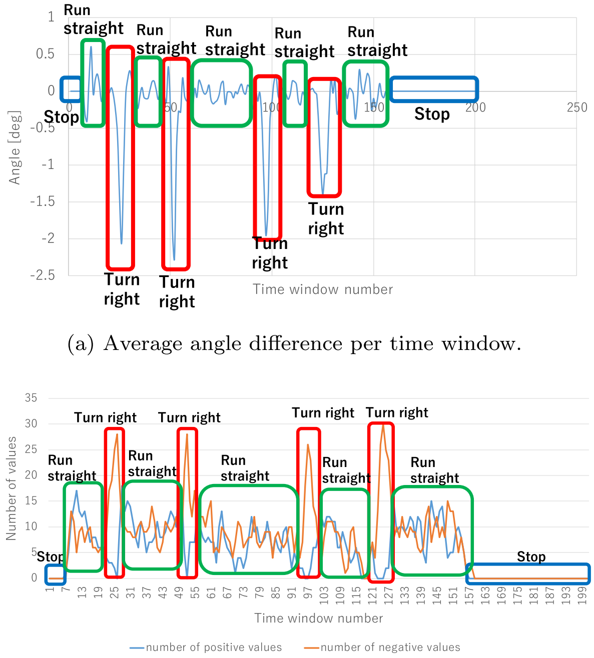
Hence, we can distinguish between the stopping state and the running state by using the average X-axis and Z-axis values. In Fig. 10(a), the red boxes show when the bicycle turns to the right. The average X-axis and Z-axis values clearly demonstrate when the bicycle turns to the right. This is mainly because the bicycle leans to the right when it turns to the right. However, when we see Fig. 10(b), the Y-axis values are always negative when we run the bicycle and cannot observe when the bicycle turns to the right.

When we ran the bicycle around the experimental route counterclockwise, the similar X-axis, Y-axis and Z-axis values were obtained from the acceleration sensor.

In summary, we can conclude that the average X-axis and Z-axis values are required to recognize the bicycle behaviors from the acceleration sensor.

(2) Gyro Sensor Values

Similarly, we obtained the Z-axis values of the gyro sensor every $t_s = 0.05$ seconds. For every time window whose size is 1.6 seconds, we calculated the angle difference value based on Eq. (3) and then they are averaged. Figure 11(a) plots the results. We also obtained the number of positive values and the number of negative values in each time window, which are plotted in Fig. 11(b). Note that every time window overlaps half to the next time window.



(a) Average angle difference per time window.

(b) The number of positive angle difference values and negative angle difference values per time window obtained from the gyro sensor.

Fig. 11 How much the gyro sensor values contribute to bicycle behaviors.

All of the average angle difference values, positive values, and negative values are much dependent on the bicycle behaviors. They become closer to zero when the bicycle stops. They become the local maximum or minimum when the bicycle turns to the right.

When we ran the bicycle around the experimental route counterclockwise, the similar angle difference values were obtained from the gyro sensor.

In summary, we can conclude that all of the average angle difference values, positive values, and negative values are required to recognize the bicycle behaviors from the gyro sensor.

(3) Feature Values for Bicycle Behavior Recognition

As described above, we set up the feature values for every time window to recognize bicycle behaviors as follows:

1. The average X-axis value obtained from the acceleration sensor
2. The average Z-axis value obtained from the acceleration sensor
3. The average angle difference value obtained from the gyro sensor
4. The number of positive angle difference values from the gyro sensor
5. The number of negative angle difference values from the gyro sensor

3.2.2 Time Window Size

Based on [21], we utilize the sliding window method and

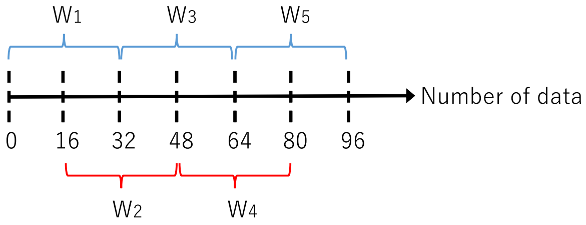


Fig. 12 Sliding window method.

Table 1 F-measure of bicycle behavior recognition when varying the time window size.

	Stop	Run straight	Turn right	Turn left
20 samples	0.99	0.98	0.86	0.75
32 samples	1.00	0.99	0.89	0.90
60 samples	1.00	0.98	0.90	0.91

extract the five feature values designed in the previous section for each time window. Now, how to set up the window size becomes the largest problem.

Firstly, in order to use much information and change the feature values smoothly in each time window, adjacent time windows should be overlapped. As in [21], every time window overlaps half to the next time window as shown in Fig. 12.

Secondly, we conducted a preliminary experiment to set up the time window size. We set the time window size such that every time window includes 20 samples, 32 samples, and 60 samples and recognized the bicycle behaviors. The other experimental conditions are the same as the ones in Sect. 4.1. We ran the bicycle around the experimental route as shown in Fig. 5 27 times counterclockwise and 28 times clockwise as the training data. We also prepared the test data in which the bicycle turns to the right 3 times and turns to the left 3 times. Table 1 summarizes the results where the F-measures of the four bicycle behaviors are shown (the definition of F-measure will be described in Sect. 4.1).

When the time window include 20 samples, the F-measure becomes lower than the others (32 samples and 60 samples). This is because if the time window size is small, it is difficult to capture the bicycle behaviors. Comparing the time window size with 32 samples and that with 60 samples, the F-measures are almost the same. However, if the time window size is too large, we cannot recognize the bicycle behavior in a real-time manner. The time window size should become small enough.

Hence, we set up the time window size such that every time window includes 32 samples, i.e., every time window has the size of 0.05s × 32 = 1.6 seconds.

3.2.3 The Classification Method

According to the discussions above, we calculate the five-dimensional feature vectors for 1.6-second time windows in training data and learn them using the random forest classifier. In the random forest classifier, we set up the number of decision trees to be 300 for obtaining stable recognition results and the number of features to be $\sqrt{5}$ which is the default recommended value.

After that, we recognize bicycle behaviors giving unknown test data by using the learned random forest classifier.

4. Evaluation Experiment

We have implemented our proposed method in Python using the scikit-learn library. We have applied it to real data and evaluated the proposed method.

4.1 Experimental Method and Condition

We use HUAWEI Mate 9 [16] as the experimental smartphone and obtain the acceleration and gyro sensor values every $t_s = 0.05$ seconds. Figure 1 shows the smartphone mounted on a bicycle. We install the smartphone on the bicycle handlebar with Y-axis directing forward and the screen upward. Then a user runs the bicycle along the experimental route as shown in Fig. 5. The experimental route is a residential district in Tokyo, Japan, with sidewalks and telegraph poles, partly sloping. As in Fig. 5, in the top side, the right point is 3m-higher than the left point. In the bottom side, the right point is 4m-higher than the left point. We prepare the training data running the bicycle on this experimental route several times counterclockwise and clockwise. After that, we calculate the five feature values for each time window from the training data and manually label them in advance. We also prepare the test data running the bicycle on other experimental route. We manually label test data and compare them with the result of bicycle behavior recognition. We use the F-measure F to evaluate the correctness of our bicycle behavior recognition, which can be calculated by:

$$F = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{4}$$

where $Recall$ refers to the recall and $Precision$ refers to the precision. The recall is defined by the ratio of positive data identified as positive over all the positive data, and the precision is defined by the ratio of the data identified as positive that is actually positive over all the data identified as positive. Since the recall and precision are in a tradeoff relationship, the F-measure is often used for integrating these values.

4.2 Effect of Noise Reduction Processes Based on Filtering

We conduct evaluation experiments to investigate the effect of the noise reduction processes on the filtering proposed in Sect. 3.1.

4.2.1 Conditions

We have prepared the training data running the bicycle on the experimental route 5 times counterclockwise and 4 times clockwise. The total number of the five-dimensional feature vectors became 1,656. We have also prepared the test data

Table 2 Bicycle behavior recognition result when noise reduction is not applied.

	Stop	Run straight	Turn right	Turn left	F-measure
Stop	30	7	0	0	0.87
Run straight	1	198	3	0	0.96
Turn right	0	2	13	0	0.84
Turn left	0	2	0	7	0.87

Table 3 Bicycle behavior recognition result when noise reduction is applied.

	Stop	Run straight	Turn right	Turn left	F-measure
Stop	34	3	0	0	0.96
Run straight	0	199	1	2	0.98
Turn right	0	1	14	0	0.93
Turn left	0	2	0	7	0.78

in which the bicycle turned to the left twice and turned to the right three times on the same experimental route. The total number of the five-dimensional feature vectors became 263. We have used the non-noise-reduced data and the noise-reduced data for comparison.

4.2.2 Experimental Result

Table 2 demonstrates the bicycle behavior recognition results when noise reduction is not applied to every sample data and Table 3 demonstrates the bicycle behavior recognition results when noise reduction is applied. In these tables, the first column shows the correct label and the first row shows the inferred label. For example, in Table 2, 30 data are correctly infered into the stop state and 7 data are mistakenly infered into the run-straight state.

The F-measures of stop, run-straight, and turn-right states become larger when applying noise reduction. The F-measure of turn left becomes larger when not applying noise reduction. This is because, as a result of applying noise reduction, the run-straight state is mistakenly recognized as the turn-left state, since the data at the beginning of turning left and the data at run-straight are close to each other.

In order to further investigate whether the noise reduction processes based on filtering is effective or not, we conduct 8-fold cross-validation. Firstly, we randomly partition the training data into eight groups. Then we pick up one of the eight groups and consider it to be test data. We consider the remaining seven groups to be training data. Secondly, we learn the seven groups and set up the random forest classifier. Finally, we classify the test data picked up first into four bicycle behaviors.

In this case, we use the accuracy to measure how much we can obtain correct answers. Let $n_{correct}$ be the number of correct answers and N be the total number of data. Then the accuracy is expressed by:

$$accuracy = \frac{n_{correct}}{N} \tag{5}$$

Since we can pick up any one of the eight groups in 8-fold cross-validation, we have eight accuracy values. Then these values are averaged and summarized in Table 4. As can be seen in this table, the accuracy improves when noise reduction is applied.

Table 4 Average accuracy in 8-fold cross-validation.

Noise reduction not applied	Noise reduction applied
0.964	0.976

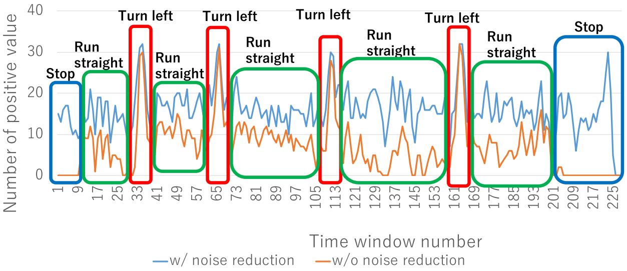


Fig. 13 Comparison of the numbers of positive angle difference values.

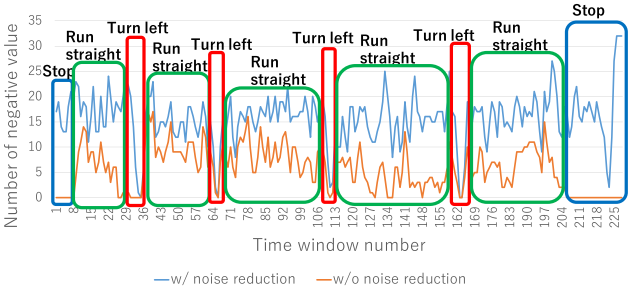


Fig. 14 Comparison of the numbers of negative angle difference values.

Particularly, Figs. 13 and 14 show the numbers of positive and negative angle difference values. These data are part of the training data prepared in the experiment of this subsection. As can be seen in these figures, the stop state and the running-straight state can be clearly distinguished when noise reduction is applied.

Overall, we can conclude that applying the noise reduction processes based on filtering is effective to recognize the bicycle behaviors.

4.3 Comparison with the Existing Method

As far as we know, there exists only one method [8] which recognizes bicycle behaviors using a smartphone mounted on a bicycle. This method uses support vector machine based machine learning and recognizes bicycle behaviors without filtering the sensor values by installing a smartphone beside the rear wheel axle of the bicycle.

In this section, we compare the proposed method to the existing method [8] and a method of installing a smartphone on a rear axle of a bicycle as in the existing method [8].

4.3.1 Setups

The setups for the three methods are as follows:

Proposed method: In the proposed method, User A and User B participated the experiment. User A has prepared the training data, running the bicycle on the experimental route 27 times counterclockwise and 28 times clockwise. The total number of the five-dimensional feature vectors became 8,969.
We have also prepared the test data as follows:

User A: User A has run his/her bicycle turning to the left 3 times and turned to the right 3 times on the same experimental route. The total number of the five-dimensional feature vectors became 295. We have used the noise-reduced data here.

User B: User B has run his/her bicycle turning to the left 4 times and turned to the right 4 times on the different experimental route. The total number of the five-dimensional feature vectors became 261. We have also used the noise-reduced data here.

Method [8]: We cite the results of the method [8] just for reference purpose[†]. The F-measures are not shown in [8] and then we calucate them based on the bicycle behavior recognition results.

Method of installing a smartphone on a rear axle of a bicycle as in [8]:
We have installed a smartphone on a rear axle of a bicycle as in the existing method [8], on which influence of noises caused by the handlebar motions can be reduced. Three-axis acceleration sensor values and three-axis gyro sensor values without particular noise reduction processes have been obtained every 0.05 seconds.
We have prepared the training data running the bicycle on the experimental route of Fig. 5 15 times counterclockwise and 15 times clockwise. The total number of the five-demensional feature vectors described in Sect. 3.2.1 became 4,557. We have also prepared the test data in which the bicycle turned to the left 4 times and turned to the right 4 times. The total number of the five-demensional feature vectors became 337. After that, we have recognized the bicycle behaviors by the random forest classifier.

4.3.2 Experimental Result

Tables 5, 6, 7 and 8 summarize the results. The meaning of each number in these tables is the same as Tables 2 and 3.

In the existing method [8], the F-measures of the accelerating state and the constant-speed state exceed 0.8, but those in the clockwise-circle-running state and the counterclockwise-circle-running state do not exceed 0.8. In the method of installing a smartphone on a rear axle of a bicycle, the F-measures of all the four states become 0.85 or higher. The proposed method realizes that the F-measures of all the four states become around 0.9 or more resulting in high accuracy, even when the smartphone is installed on the noisy bicycle handlebar. Futhermore, the proposed method realizes high accuracy even if the test user is different from the training user.

Comparing the existing method [8] and the proposed method, we can see that the proposed method has higher F-measures, particularly in right turn and left turn, although we cannot directly compare the clockwise-circle-running state to the trun-right state and the counterclockwise-circle-running state to the turn-left state. By installing the smartphone on the bicycle handlebar with Y-axis directing forward and the screen upward, the gyro sensor values introducing the nosie reduction process correctly give the bicycle turning state.

Comparing the method of installing the smartphone on the rear axle of the bicycle and the proposed method, in all the four states, the recognition results are almost the same as Tables 5 and 6. These results clearly indicate that, even if the smartphone is installed on the bicycle handlebar, we can recognize the bicycle behaviors with almost the same accuracy as the one when a smartphone is installed on a rear axle of a bicycle. By installing the smartphone on the bicycle handlebar, a user can easily use his/her smartphone on the bicycle and our method enables that.

Note that the F-measure in the right/left-turn state of our method is slightly lower than the other states. This is because the feature values are not much changed at the beginning of turning right/left.

In summary, the proposed method realizes that, even if (1) a smartphone is installed on a bicycle handlebar that is easy to see from a user but has periodic noises, high accuracy of the bicycle behavior recognition is achieved by using (2) the noise reduction processes based on filtering.

As for (2), Sect. 4.2 discusses whether the noise reduction processes based on filtering is effective or not, assuming that (1) the smartphone is installed on a bicycle handlebar that is easy to see from the user but has periodic noises. Section 4.2 actually confirms that (2) the noise reduction processes based on filtering is effective.

As for (1), we cannot compare our proposed method directly to [8]. Hence, we have conducted the further experiment in order to investigate whether proposed method is effective or not from the viewpoint of (1). The results

[†] According to [8], the accuracy of bicycle behavior recognition using the support vector machine method (SVM method) is higher than the nearest neighbor method (NN method). Since the result on the SVM method is superior to the NN method, we cited the experimental results on the SVM method from [8].

Table 5 Bicycle behavior recognition results of User A (ours).

	Stop	Run straight	Turn right	Turn left	F-measure
Stop	32	0	0	0	1.00
Run straight	0	232	0	1	0.99
Turn right	0	3	12	0	0.89
Turn left	0	2	0	13	0.90

Table 6 Bicycle behavior recognition results of User B (ours).

	Stop	Run straight	Turn right	Turn left	F-measure
Stop	40	0	0	0	1.00
Run straight	0	180	1	0	0.98
Turn right	0	3	18	0	0.90
Turn left	0	2	0	17	0.94

Table 7 Bicycle behavior recognition results of the existing method cited from [8].

	Accelerating	Constant speed	Running clockwise circle	Running counterclockwise circle	F-measure
Accelerating	65	7	1	1	0.88
Constant speed	6	41	0	0	0.86
Running clockwise circle	1	0	40	30	0.62
Running counterclockwise circle	0	0	17	63	0.72

Table 8 Bicycle behavior recognition results (a smartphone is installed on the rear axle of the bicycle).

	Stop	Run straight	Turn right	Turn left	F-measure
Stop	39	7	0	0	0.91
Run straight	1	244	1	2	0.96
Turn right	0	6	20	0	0.85
Turn left	0	2	0	12	0.85

above indicate that our proposed method can recognize the bicycle behaviors with almost the same accuracy as the one when a smartphone is installed on a rear axle of a bicycle, on which influence of noises caused by the handlebar motions can be much reduced. Further, the bicycle behaviors by the proposed method does not much depend on users.

5. Discussions

5.1 Application Scenario

We can consider the application scenario for predicting dangerous riding behaviors using our proposed method as follows:

Dangerous bicycle behaviors refer to the way to ride the bicycle that may cause bicycle accidents [22]. According to [22], [23], one of such dangerous riding behaviors is strong meandering of bicycles, for example. Normal riding behaviors refer to the bicycle riding behaviors that are not the dangerous riding behaviors.

When we recognize right-turn/left-turn states alternatively in a very short period of time or we recognize stop/run-straight states alternatively in a very short period of time using our proposed method, we consider the bicycle behavior to be strong meandering of bicycles, which may cause bicycle accidents. When we recognize one of the stop/run-straight/right-turn/left-turn states in some amount of time, we consider the bicycle behavior not to be strong meandering, which must be a normal riding behavior.

When strong meandering of bicycles are recognized and they continue in some amount of time, we can predict that dangerous bicycle riding behaviors will continue in the future and bicycle accidents may occur due to them.

Note that, there must be dangerous riding behaviors other than the one above. By recognizeing dangerous bicycle riding behaviors similarly, we expect that we can also predict bicycle accidents caused by those dangerous bicycle riding behaviors.

How to really predict dangerous riding behaviors and evaluate them are the important future works.

5.2 The Installation Position Offset of a Smartphone

Generally, a bicycle user uses the smartphone holder such as in [26] when installing a smartphone on a bicycle. The user cannot usually install the smartphone exactly on the top of the stem and the user installs the smartphone on the position slightly off to the left or to the right of the bicyle handlebar center. Therefore, we install the smartphone on the right side of the bicycle handlebar in this paper as in Fig. 1.

In this subsection, we investigate how much the installation position offset (right offset or left offset to the bicycle handlebar center) of a smartphone will affect the bicycle behavior recognition results of our proposed method.

Firstly, when we install the smartphone on the right side of the bicycle handlebar, the bicycle behavior recognition result is shown in Table 5, whose F-measures become 0.89–1.00.

Table 9 Bicycle behavior recognition results (a smartphone is installed on the left side of the handlebar).

	Stop	Run straight	Turn right	Turn left	F-measure
Stop	33	0	0	0	0.95
Run straight	3	238	0	1	0.99
Turn right	0	2	18	0	0.95
Turn left	0	1	0	15	0.94

Secondly, we conduct an experiment installing the smartphone on position slightly off to the left of the bicycle handlebar center. In this case, we have also prepared the test data in which User A ran the bicycle turning to the left 4 times and turning to the right 4 times on the experimental route of Fig. 5. The total number of the five-dimensional feature vectors became 311. The training data used is the same as the one used in Table 5, i.e., the training data have been obtained installing the smartphone on the right side of the bicycle handlebar.

Table 9 demonstrates the bicycle behavior recognition results when the smartphone is installed on the left side of the bicycle handlebar. The meaning of each number in this table is the same as Tables 2 and 3. The F-measures of all the four states exceeded 0.90 and are almost the same as the bicycle behavior recognition results when the smartphone is installed on the right side of the bicycle handlebar. Therefore, we can conclude that the installation position offset of a smartphone almost does not affect the bicycle behavior recognition.

5.3 Slope

The bicycle traffic accidents can be more likely to occur due to abrupt acceleration and/or turning on slopes. Now we investigate how much slopes affect the bicycle behavior recognition results of our proposed method.

Firstly, the gravitational acceleration is removed beforehand from the acceleration values and hence the effect of slope must be small enough in the acceleration values.

Secondly, since the smartphone shown in Fig. 1 runs parallel to the ground even while the bicycle runs on a slope, the effect of slope to the gyro sensor must be also small enough.

The experimental route used in the experiments is a residential district in Tokyo, Japan which includes gentle slopes. As shown in Fig. 5, the experimental route includes two gentle slopes. As Table 5 demonstrates, our proposed method accurately recognizes the bicycle behaviors in the experiments while the bicycle runs on slopes.

How accurate the proposed method can recognize bicycle behaviors on steep slopes is also the important future works. For example, we can use the quaternion calculation method [24], [25] in this case.

6. Conclusion

In this paper, we have proposed a bicycle behavior recognition method using three-axis acceleration sensor and three-axis gyro sensor equipped with a smartphone. In the proposed method, focusing on the periodic handlebar motions for balancing while running a bicycle, we reduce the noises caused by them by effectively applying noise reduction processes based on filtering. After that, we use a machine learning approach for recognizing the bicycle behaviors, utilizing the five-dimensional features in bicycle behavior recognition. The experimental results demonstrate that the proposed method accurately recognizes the four bicycle behaviors of stop, run straight, turn right, and turn left and its F-measure becomes around 0.9 in all the cases. Even if the smartphone is installed on the bicycle handlebar, we can recognize the bicycle behaviors with the same accuracy as the one when a smartphone is installed on a rear axle of a bicycle. In the future, we will develop a method to recognize and predict dangerous riding behaviors such as strong meandering.

References

[1] T. Kamimura, T. Kitani, and T. Watanabe, "A system to comprehend a motorcycle's behavior using the acceleration and gyro sensors on a smartphone," Global Research and Education, 2012.

[2] T. Kitani, "Bikeinformatics: A concept of the study of its for two-wheel vehicles with information science and technologies," Proc. IPSJ Multimedia, Distributed, Cooperative and Mobile Symposium (DICOMO2013), pp.1517–1524, 2013 (in Japanese).

[3] T. Kamimura, T. Kitani, and D.L. Kovacs, "Automatic classification of motorcycle motion sensing data," Proc. 2014 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp.145–146, 2014.

[4] N.P. Agency, http://www.npa.go.jp/news/release/index.html/

[5] N.P. Agency, "Definition of a bicycle," http://www.keishicho.metro.tokyo.jp/bicyclette/jmp/bicyclette.pdf

[6] S. Smaldone, C. Tonde, V.K. Ananthanarayanan, A. Elgammal, and L. Iftode, "Improving bicycle safety through automated real-time vehicle detection," Department of Computer Science, Rutgers University, vol.110, 2010.

[7] S.B. Eisenman, E. Miluzzo, N.D. Lane, R.A. Peterson, G.S. Ahn, and A.T. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," ACM Trans. Sen. Netw. (TOSN), vol.6, no.1, pp.1–39, 2009.

[8] H. Goto, Behavior Recognition of the Bicycle using the Acceleration Sensor (in japanese), http://ist.mns.kyutech.ac.jp/miura/papers/master2013-goto.pdf

[9] H. Niki and T. Murakami, "An approach to self stabilization of bicycle motion by handle controller," IEEJ Trans. IA, vol.125, no.8, pp.779–785, 2005.

[10] T. Saguchi, K. Yoshida, and M. Takahashi, "Stable running control of autonomous bicycle robot," Trans. Japan Society of Mechanical Engineers Series C, vol.73, no.731, pp.2036–2041, 2007.

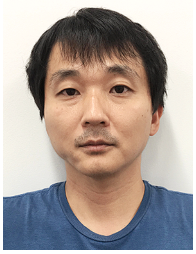
[11] H. Yamanaka, P. Xiaodong, and J. Sanada, "Evaluation models for cycliststf perception using probe bicycle system," J. Eastern Asia Society for Transportation Studies, vol.10, pp.1413–1425, 2013.

[12] Y. Usami, K. Ishikawa, T. Takayama, M. Yanagisawa, and N. Togawa, "Bicycle behavior recognition using 3-axis acceleration sensor and 3-axis gyro sensor equipped with smartphone," Proc. IEEE Interna-

tional Conference on Consumer Electronics in Berlin (ICCE-Berlin), 2018.
[13] M. Hoffmann, M. Mock, and M. May, "Road-quality classification and bump detection with bicycle-mounted smartphones," Proc. the 3rd International Conference on Ubiquitous Data Mining, pp.39–43, 2013.
[14] H.J. Lee and S. Jung, "Gyro sensor drift compensation by Kalman filter to control a mobile inverted pendulum robot system," Proc. IEEE International Conference on Industrial Technology (ICIT), 2009, pp.1–6, 2009.
[15] H. Goto and M. Miura, "Examination of sensor positions to detect bicycle speeding behavior," Proc. KES International Coference on Intelligent Interractive Multimedia: System and Service (KES-IIMSS), pp.204–211, 2013.
[16] HUAWEI, "HUAWEI Mate 9," https://consumer.huawei.com/jp/phones/mate9/
[17] L. Learning, "Rotation angle and angular velocity|physics," https://courses.lumenlearning.com/physics/chapter/6-1-rotation-angle-and-angular-velocity/
[18] L.E. Peterson, "K-nearest neighbor," Scholarpedia, vol.4, no.2, p.1883, 2009.
[19] M.M. Adankon and M. Cheriet, Support Vector Machine, pp.1303–1308, Springer, 2009.
[20] L. Breiman, "Random forests," Mach. Learn., vol.45, no.1, pp.5–32, 2001.
[21] L. Bao and S.S. Intille, "Activity recognition from user-annotated acceleration data," Pervasive Computing, pp.1–17, 2004.
[22] W. Gu, Z. Zhou, Y. Zhou, H. Zou, Y. Liu, C.J. Spanos, and L. Zhang, "BikeMate: Bike riding behavior monitoring with smartphones," Proc. 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2017.
[23] B. Hartung, N. Mindiashvili, R. Maatz, H. Schwender, E.H. Roth, S. Ritz-Timme, J. Moody, A. Malczyk, and T. Daldrup, "Regarding the fitness to ride a bicycle under the acute influence of alcohol," Int. J. Legal Med., vol.129, no.3, pp.471–480, 2015.
[24] K.W. Spring, "Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: a review," Mech. Mach. Theory, vol.21, no.5, pp.365–373, 1986.
[25] J. Favre, B. Jolles, O. Siegrist, and K. Aminian, "Quaternion-based fusion of gyroscopes and accelerometers to improve 3D angle measurement," Electron. Lett., vol.42, no.11, pp.612–614, 2006.
[26] Vibrelli, http://vibrelli.com/products/

Yuri Usami received the B. Eng. degree from Waseda University in 2017 in computer science and engineering, where she is working towards M. Eng. degree. Her research interests include intelligent transportation system, especially analysis of human mobility.

Kazuaki Ishikawa received the M. Eng. degree from Tokyo University of Science in 2008 in electrical engineering, He is presently working in the ZENRIN DataCom Co., Ltd.



Toshinori Takayama received the bachelor degree from Yokohama City University in 1998 in Philosophy. He is presently working in the ZENRIN DataCom Co., Ltd.

Masao Yanagisawa received the B. Eng., M. Eng., and Dr. Eng. degrees from Waseda University in 1981, 1983, and 1986, respectively, all in electrical engineering. He was with University of California, Berkeley from 1986 through 1987. In 1987, he joined Takushoku University. In 1991, he left Takushoku University and joined Waseda University, where he is presently a Professor in the Department of Computer Science and Communications Engineering. His research interests are combinatorics and graph theory, computational geometry, VLSI design and verification, and network analysis and design. He is a member of IEEE, ACM, and IPSJ.

Nozomu Togawa received the B. Eng., M. Eng., and Dr. Eng. degrees from Waseda University in 1992, 1994, and 1997, respectively, all in electrical engineering. He is presently a Professor in the Department of Computer Science and Communications Engineering, Waseda University. His research interests are intelligent transportation system, ingtrated system design, graph theory, and computational geometry. He is a member of IEEE and IPSJ.