2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin).

Bicycle Behavior Recognition using Sensors Equipped with Smartphone

Yuri Usami*†, Kazuaki Ishikawa‡, Toshinori Takayama‡, Masao Yanagisawa* and Nozomu Togawa*†

*Dept. of Computer Science and Communications Engineering, Waseda University
†Email: {yuri.usami, togawa}@togawa.cs.waseda.ac.jp
‡Zenrin DataCom Co., LTD.

***Abstract*—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. 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 while the F-measure of the existing method just reaches 0.6–0.8.**

***Index Terms*—behavior recognition, smartphone, acceleration sensor, gyro sensor, bicycle.**

I. Introduction

A. Bicycle behavior recognition

It becomes possible to prevent accidents beforehand by predicting dangerous riding behavior based on recognition of automobile/motorcycle behaviors [1]. On the other hand, according to the bicycle accident analysis data from the National Police Agency in Japan [2], the number of bicycle accidents in Tokyo area became 11,901 in 2017, which occupies around one third of all the traffic accidents in Tokyo. In order to reduce the bicycle accidents, it is quite necessary to recognize bicycle behaviors and prevent dangerous riding beforehand [3, 4] 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] 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. [5].

B. 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 [3]:**
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 [3]. The method requires the dedicated sensor devices on a bicycle and thus they require many costs to recognize bicycle behaviors.
2) **Bicycle behavior recognition methods using a smartphone-mounted sensors [4]:**
In [4], 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, when turning right/left at a slow speed, the angular velocity change becomes too much smaller and then it is hard to distinguish between straight-run states and right/left-turn states at that time. Furthermore, users cannot use smartphones even while stopping.

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

C. 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.

D. 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

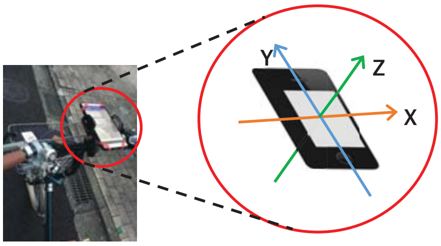


Fig. 1: The axies of the acceleration/gyro sensors equipped with a smartphone.

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 samples and test data composed of 295 samples. 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 F-measures in turn right and turn left are just 0.62 and 0.72 in the existing method [4], respectively, whereas those in our proposed method are 0.89 and 0.90, respectively, which means that our proposed method outperforms the conventional state-of-the-art method.

remarks.

II. 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 gyro three-axis sensor is used. The axies of the acceleration sensor and gyro sensor equipped with a smartphone is shown in Fig. 1. The three-axis acceleration sensor can obtain the X-axis, Y-axis, and Z-axis acceleration values. Assume that the smartphone is installed on the bicycle with the Y-axis directing forward and the screen upward. 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 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 Then, the bicycle behavior recognition problem is defined as follows:

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.*

III. 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 [5, 6]:

(A) Noises from road surface
(B) Noises due to handlebar motions
(C) Noises due to drifts

(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.

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 [7]. Thus, in [4], 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.

A. Noise reduction processes based on filtering

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

1) Filtering for three-axis acceleration sensor: The noises of (A) and (B) affect the three-axis acceleration sensor. From the preliminary experiment result, we can find that 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. 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 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 Section III-B2). By averaging the 32 values in each time window, we can obtain the average X-axis value in every time window.

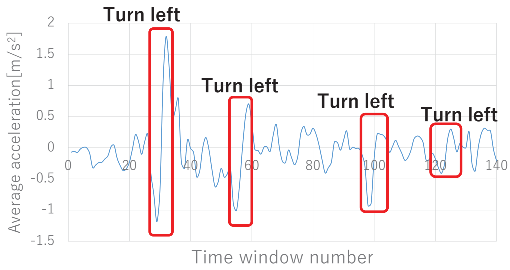
Fig. 4 shows the average X-axis acceleration values for time windows, where the bicycle turns to the left at the time window numbers of 29–33, 56–59, 99–103, and 122–126. Before applying the LPF (see Fig. 2a), we can observe the local maximum and minimum values at the time window numbers of 29–33, 56–59, and 99–103 but we cannot see them at the time window numbers of 122–126. This is because the bicycle turns to the left at the low speed there. On the other hand, after applying the LPF (see Fig. 2b), we can successfully observe the local maximum and minimum values at the time window numbers of 29–33, 56–59, 99–103, and 122–126. 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.

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. Hence we use the Z-axis gyro sensor values to recognize bicycle behaviors.

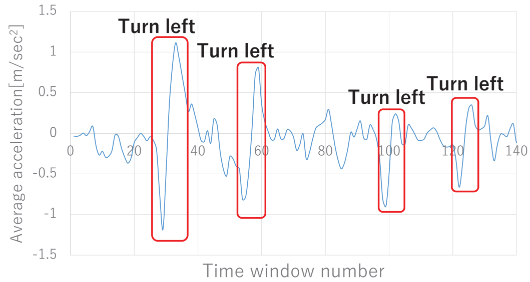
Now we conducted a preliminary experiment to investigate how much Z-axis values of the gyro sensor are affected by (C) drift noises when locating the smartphone on a horizontal desk with its screen upward. The gyro sensor obtained the Z-axis angular velocity $\omega_z(i)$ $(i = 0, 1, 2, \cdots)$ every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

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

Fig. 3 shows the angle values before noise reduction and after noise reduction. When we just located the smartphone on a horizontal desk with the screen upward, the angle was gradually increased. 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.



(a) Before noise reduction.



(b) After noise reduction.

Fig. 2: Average X-axis acceleration value per time window (time window size = 32).

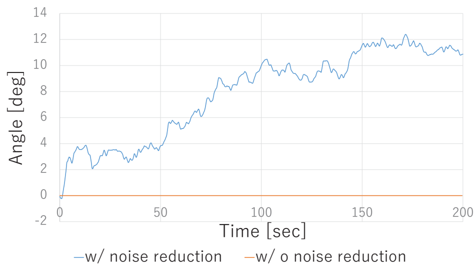
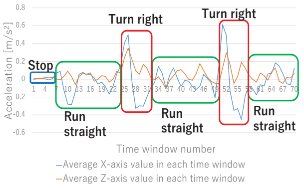


Fig. 3: Drift of Z-axis in the gyro sensor.

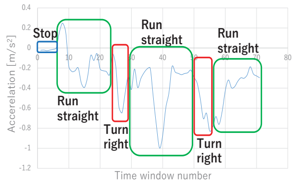
Since the bicycle is balanced by keeping its handlebar periodically turning left or right, we can set up the threshold value d_{th} so that, if $d_z(i)$ becomes larger than d_{th}, we can consider it that the bicycle actually changes its direction. However, if $d_z(i)$ does not become larger than d_{th}, 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:

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

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 $d_{th} = 0.49$.



(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. 4: How much the acceleration sensor values contribute to bicycle behaviors.

When we apply Eq. (2) to Z-axis values of the gyro sensor, we can obtain the flat wave form as depicted in Fig. 3, where all the noises of (A), (B), and (C) are reduced.

B. 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.

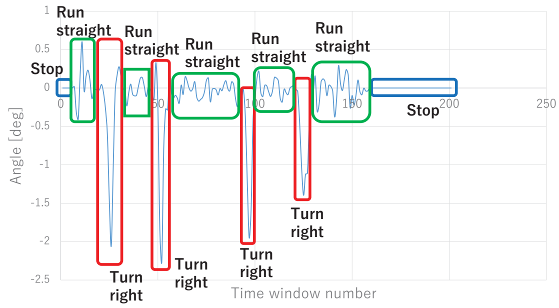
There have been proposed many machine learning approaches such as a k-nearest neighborhood method and a support vector machine method . However, bicycle behaviors should be correctly recognized by the values from the acceleration sensor and gyro sensor, even if some of them include the outlier values. Hence, we select the random forest classifier [8], 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.

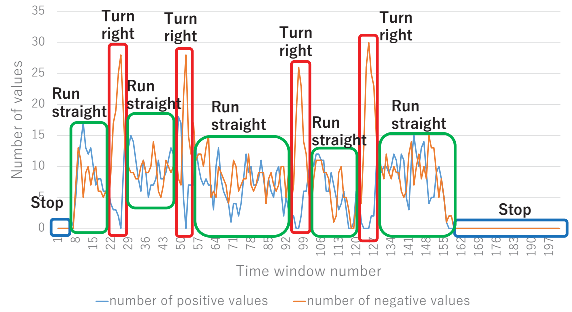
1) Bicycle behavior features: Through several preliminary experiments, we observe which ones of the acceleration sensor values and gyro sensor values most contribute to classify the bicycle behaviors.

a) Acceleration sensor values: We installed a smartphone on the bicycle handlebar with its Y-axis directing forward and ran the bicycle around the experimental route 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. 4a and 4b. 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. Hence, we can distinguish between the stopping state and the running state by using the average X-axis and Z-axis values. In Fig. 4a, 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. 4b, the Y-axis values are always negative when we run the bicycle and cannot observe when the bicycle turns to the right.



(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. 5: How much the gyro sensor values contribute to bicycle behaviors.

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.

b) 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. (2) and then they are averaged. Fig. 5a 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. 5b. Note that every time window overlaps half to the next time window.

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.

c) 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:

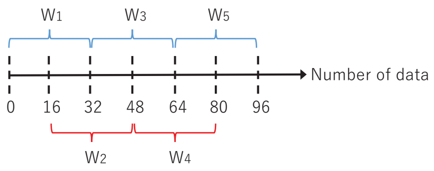


Fig. 6: Sliding window method.

TABLE I: 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

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

2) Time window size: We utilize the sliding window method and 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. Every time window overlaps half to the next time window as shown in Fig 6.

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 Section IV-A. We ran the bicycle around the experimental route 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 I summarizes the results where the F-measures of the four bicycle behaviors are shown (the definition of F-measure will be described in Section IV-A).

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 $0.05\text{s} \times 32 = 1.6$ seconds.

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.

Fig. 7: Comparison of the numbers of positive angle difference values.

IV. 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.

A. Experimental method and condition

We use HUAWEI Mate 9 [9] as the experimental smartphone and obtain the acceleration and gyro sensor values every $t_s = 0.05$ seconds. 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. The experimental route is a residential district in Tokyo, Japan, with sidewalks and telegraph poles, partly sloping. 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:

$$F = \frac{2 \times Recall \times Precision}{Recall + Precision} \qquad (3)$$

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.

B. Comparison with the existing method

As far as we know, there exists only one method [4] which recognizes bicycle behaviors using a smartphone mounted on a bicycle. This method uses support vector machine based

TABLE II: Bicycle behavior recognition results of the existing method cited from [4].

	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

Fig. 8: Comparison of the numbers of negative angle difference values.

TABLE III: Bicycle behavior recognition results (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

machine learning and recognizes bicycle behaviors without filtering the sensor values by installing a smartphone beside the rear wheel axle of the bicycle. For comparison purpose, we pick up the method [4] and compare the results.

1) Experimental conditions: In the proposed method, we have 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 in which the bicycle turned to the left three times and turned to the right three 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.

In the comparison method, we have cited the recognition results from [4].

2) Experimental result: Tables II and III summarize the results. In the existing method [4], 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 and thus the accuracy becomes low. On the other hand, the proposed method realizes that the F-measures of all the four states become around 0.9 or more resulting in high accuracy.

Comparing the existing method [4] and the proposed method, we can see that the proposed method has higher F-measures in right turn and left turn. This is because, 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.

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, it can be seen that the proposed method recognizes the bicycle behaviors more accurately than the existing method.

V. 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, while the F-measure of the existing method just reaches 0.6–0.8.

In the future, we will develop a method to recognize dangerous riding such as abrupt steering and abrupt braking, other than the normal riding states.

Acknowledgments

This research is supported in part by Waseda University Grant for Special Research Projects (Nos. 2018K-189 and 2018B-088).

References

[1] Tsukasa Kamimura, Tomoya Kitani, and Takashi Watanabe, “A system to comprehend a motorcycle’s behavior using the acceleration and gyro sensors on a smartphone,” *Global Research and Education*, 2012.

[2] National Police Agency, http://www.npa.go.jp/news/release/index.html/.

[3] Stephen Smaldone, Chetan Tonde, Vancheswaran K Ananthanarayanan, Ahmed Elgammal, and Liviu Iftode, “Improving bicycle safety through automated real-time vehicle detection,” *Department of Computer Science, Rutgers University*, vol. 110, 2010.

[4] Hidenobu Goto and Motoki Miura, “Behavior recognition of the bicycle using the acceleration sensor (in japanese),” *IPSJ Interaction 2014*, vol. 2014, pp. 309–312, 2014.

[5] Hiroshi Niki and Toshiyuki Murakami, “An approach to self stabilization of bicycle motion by handle controller,” *IEEJ Transactions on Industry Applications*, vol. 125, no. 8, pp. 779–785, 2005.

[6] Marius Hoffmann, Michael Mock, and Michael May, “Road-quality classification and bump detection with bicycle-mounted smartphones,” in *Proc. the 3rd International Conference on Ubiquitous Data Mining*, 2013, pp. 39–43.

[7] Hidenobu Goto and Motoki 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.

[8] Leo Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] HUAWEI, “HUAWEI Mate 9,” https://consumer.huawei.com/jp/phones/mate9/.