

Pedestrian Action Recognition Using Motion Sensor and k-NN Classifier

Ningping Sun^{1,*} and Toru Tsuruoka²

¹Dept. of Human-Oriented Information System Engineering, National Institute of Technology, Kumamoto College, 2659-2 Suya, Koshi, Kumamoto 861-1102, Japan

²Advanced Electronics and Information Systems Engineering Course, National Institute of Technology, Kumamoto College, 2659-2 Suya, Koshi, Kumamoto 861-1102, Japan

*Corresponding author

Abstract—We focused on the recognition of pedestrian action using motion sensor and machine learning. In this paper, we present our method how to define feature of actions from acquired data, and how to use k nearest neighborhood classifier to data learning and evaluation.

Keywords—motion sensor; action recognition; machine learning; feature definition; k nearest neighborhood classifier

I. INTRODUCTION

Walking, stepping, and running constitute a series of basic human movement [1]. How to express these movements in computer graphics faithfully is an important task, and it is an important subject for artificial intelligence to recognize the pedestrian movement status correctly. The remarkable motion capture techniques, from large specialized systems to small simplified motion capture devices, help us to model human movement and action more efficiently and precisely before representing and analyzing them. However, the vision-based sensors used in these systems, such as cameras and depth sensors, can only capture narrower ranges in front of the sensor, making it difficult to capture various kinds of pedestrian movement both outdoors and in nature[2]. Therefore, we have introduced and employed wearable motion sensors such as acceleration sensor and angular velocity sensor into our mobile motion capture system instead of vision-based sensors.

In this study, using a group of compact wireless multifunction sensors we track pedestrian movement and detect its feature for our representation, analysis and recognition. While obtaining acceleration and angular velocity from sensors, we visualize and render leg motion with three-dimensional computer graphics [3]. After the visualization was successful, we focused on the recognition of pedestrian action using motion sensor data and machine learning. In this paper, we present our method how to define feature of actions from acquired data, and how to use k-NN classifier to data learning and evaluation.

II. THE PROPOSED APPROACH AND ALGORITHM

A. Motion Capturing with Motion Sensor TSND121

In our experiments, we attached and fixed four ATR-Promotions six-axis inertial sensors TSND121 on both legs of the tester to obtain walking acceleration and angular velocity, as

shown in Figure I. TSND121 can transmit data via Bluetooth, the acquiring range of acceleration is ± 16 G and the range of angular velocity is ± 2000 dps. When the sampling time is set to one milliseconds, we can get about 1,000 pieces of six-axis information per second. TSND121 uses right-hand coordinate system, and z points to the direction of progress.

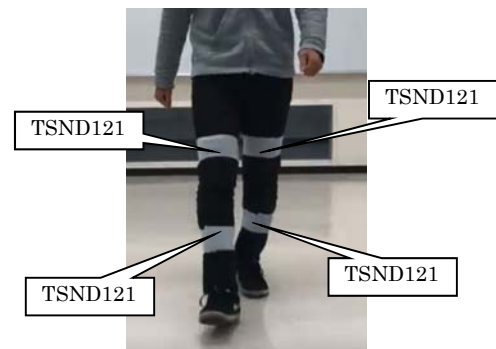


FIGURE I. WALKING EXPERIMENT WITH TSND121 ATTACHED

In many cases, the information acquired by the motion sensor might involve errors like the blue line of Figure II due to noise even when the sensor is at a stationary state. In order to improve the measurement accuracy, we reduce the errors of acceleration and angular velocity by a low pass filter (1), where $data_i$ is the uncorrected data, S_i is the data corrected, and the correction factor, $a = 0.992$. We can get the orange line of FIGURE II, which is acceleration z after corrected by (1).

$$\begin{cases} S_i = data_i, & i = 0 \\ S_i = aS_{i-1} + (1-a)data_i, & i > 0 \end{cases} \quad (1)$$

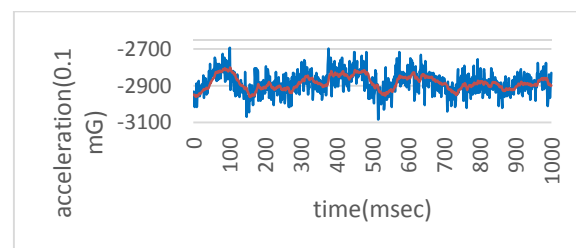


FIGURE II. THE ERROR CORRECTION OF ACCELERATION Z

In Figure III, IV and V we present samples of walking, stepping and running, and each responding six-axis information included acceleration(x, y, z) and angular velocity (x, y, z), in which we only show the data acquired from the right thigh.

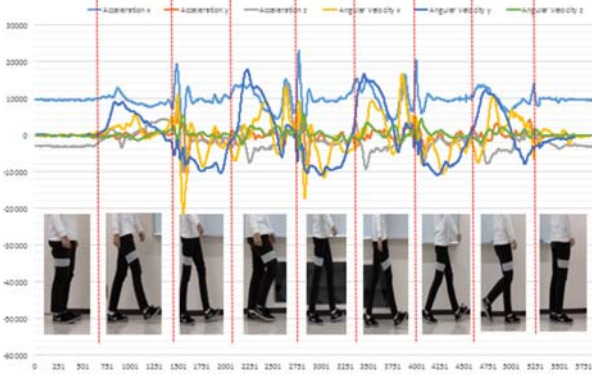


FIGURE III. WALKING SAMPLING AND ITS 6-AXIS INFORMATION

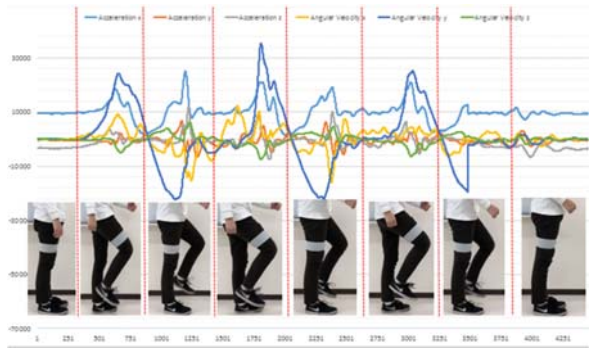


FIGURE IV. STEPPING SAMPLING AND ITS 6-AXIS INFORMATION

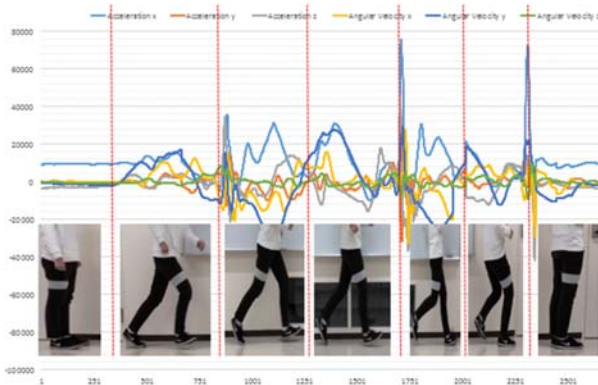


FIGURE V. RUNNING SAMPLING AND ITS 6-AXIS INFORMATION

We want to get 9000 data for our experiment, for each action such as walking, stepping and running, we sampled three times in total, three seconds each. However, related to the Bluetooth communication efficiency between the computer and the motion sensor, the number of the acquired data is less or more than 9000.

B. Feature Definition of Pedestrian Actions

The feature is necessary for forming a machine learning model. We specified the data, which is more useful for behavior recognition, as the feature of the action.

1) *Acceleration and angular velocity*: Since the three actions are forward movements, only the acceleration z is adopted from the acceleration (x, y, z). From FIGURE VI we can see that each waveform of the angular velocities(x, y, z) obtained from the motion sensor are different, so they are all adopted.

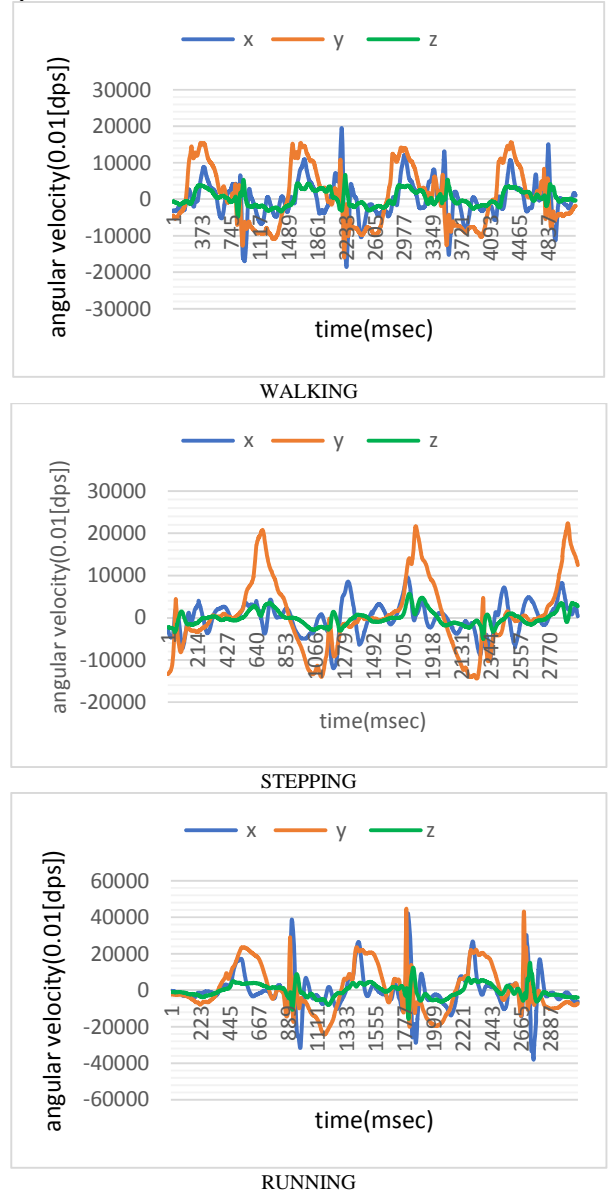


FIGURE VI. ANGULAR VELOCITY OF PEDESTRIAN ACTIONS

2) *The calculation of SDMM*: We calculate the standard deviation, maximum, and minimum of the acceleration z and angular velocities(x, y, z). These three kinds of data are defined as the feature value in our analysis, abbreviated as *SDMM*.

First, we sectioned the angular velocity data of 3 seconds into some 700 milliseconds intervals. The peak waveform at angular velocity y of Figure VI tells us that one can take one-step at about 700 milliseconds. Therefore, we divide the sampling interval by 700 milliseconds.

We divide this section into two further, and make two kinds of windows of 0 to 350 milliseconds section and 350 to 700 milliseconds section. In doing so, it is possible to divide the part where the right thigh of one-step rises and the part where the leg descends as separate windows. We calculate SDMM for each of these two divided windows, and combine these two partial SDMM to the final feature SDMM of 24-dimensional vector, which is a row of the training dataset or the test dataset. Next, we shift the window by 50 milliseconds from the previous window to next 700 milliseconds section sequentially. Similarly, we calculate SDMM for this new window. We use n as the total number of the rows of the training dataset or the test dataset.

$$n = ((\text{length}(\text{data}) - \text{length}(\text{window}))/50) \quad (2)$$

For example, when acquiring training data for 3 seconds, for the 3000 groups of acceleration z and angular velocity(x, y, z), there will be about 46 rows of SDMMs, each has 24 column, so that 1104 vectors should be calculated at most.

C. Action Recognition Using k -NN Classifier

A lazy learner classifier called k nearest neighborhood, or in short, we call it k -NN, is a simplified and non-parametric machine learning method that is important and good for the dataset which has noise or which has lot of variability. k -NN is used for classification the input consists of the k closest training example and the output is the class membership. The majority of its neighbor classifies an object. A positive integer typically small, k , refers to the number of neighbors, which we will consider and it is. Prediction based on neighborhood of test data.

We prepared the training datasets and test datasets shown in TABLE I and made predictions to test what the examples are, i.e., we want to find out how close the test action type is to already given training types. First, using (3) we calculate Euclidean distance between the test dataset and each training dataset.

TABLE I. DATASET FOR EVALUATION EXPERIMENT

dataset		Number of data (6-axis)	n
Training dataset	walking	9169	170
	stepping	9055	167
	running	8893	164
Test dataset	walking	3208	51
	stepping	3014	49
	running	3111	49

$$d_i = \sqrt{\sum_{j=1}^{24} (\text{training.SDMM}_{i,j} - \text{test.SDMM}_j)^2} \quad (3)$$

where i indicates the i th row of the dataset, $1 \leq i \leq n$, and j indicates the j th column of dataset, $1 \leq j \leq 24$.

We give three results of (3) in Figure VII, VIII and IX. In these scatter plot one small point denotes a distance between the feature SDMM of the given test action and the SDMM of the training actions. We can rank the distances according to the minimum distance. When estimating labels of test data, we use k -NN to majority vote its k immediate neighbors. Like a general way, we set $k = 3$. In Figure VII, for the blue test walking, the three nearest neighbors all are blue walking, and similarly, in Figure VIII the test red stepping matches the three of training red stepping, in Figure IX the test green running is closest to the three training green running. The results tell us SDMMs come from the same action show stronger similarity and better recognition precision.



FIGURE VII. THE RESULT OF (3) FOR THE TEST WALKING SDMM51



FIGURE VIII. THE RESULT OF (3) FOR THE TEST STEPPING SDMM49



FIGURE IX. THE RESULT OF (3) FOR THE TEST RUNNING SDMM49

Table II gives the results of our evaluation for every test SDMM. The precision and recall reach 100% in the example.

TABLE II. RESULTS OF RECOGNITION TEST

Training Dataset		The Results of Recognition Test			
		walking	stepping	running	Recall
$k = 3$	walking	51	0	0	1
	stepping	0	49	0	1
	running	0	0	49	1
	Precision	1	1	1	-
$k = 5$	walking	51	0	0	1
	stepping	0	49	0	1
	running	0	0	49	1
	Precision	1	1	1	-

III. EVALUATION AND IMPROVEMENT OF FEATURE DEFINITION

As mentioned above, the 24-dimensional vector contained in SDMM contributed to providing very high accuracy of recognition. The problem is that the amount of data is too much for real-time calculations. We searched for the possibility of reducing the amount of features while maintaining the recognition accuracy.

From Figure VI we can see that the angular velocity y plays an important role in all pedestrian behaviors. In walking, stepping, and running, angular velocity y indicates the upward and downward angular velocity in the vertical direction. As an alternative, we defined a new SDMM as the standard deviation, maximum, and minimum of the acceleration z and angular velocity y , which becomes 12-dimensional vector.

Using the data of Table I, we did the same calculation and recognition as section II. We give the results of recognition test in Table III and the distances obtained by (3) in Figure X, XI, and XII.

TABLE III. RESULTS OF RECOGNITION TEST

Training Dataset		The Results of Recognition Test			
		walking	stepping	running	Recall
$k = 3$	walking	51	0	0	1
	stepping	0	49	0	1
	running	0	0	49	1
	Precision	1	1	1	-
$k = 5$	walking	51	0	0	1
	stepping	0	49	0	1
	running	0	0	49	1
	Precision	1	1	1	-

From the results we can see the new feature *SDMM* provides a very good accuracy of recognition as well. The definition of feature depends on behavioral characteristics, for a general pedestrian action like we show in FIGURE III, IV, and V, the new *SDMM* is not bad as one option.



FIGURE X. THE RESULT OF (3) FOR THE TEST WALKING SDMM51



FIGURE XI. THE RESULT OF (3) FOR THE TEST STEPPING SDMM49

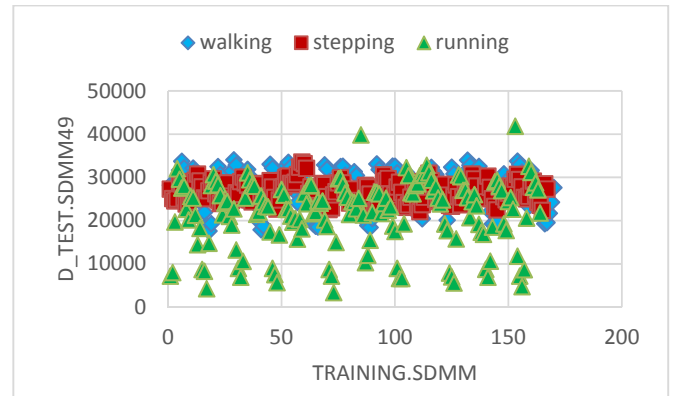


FIGURE XII. THE RESULT OF (3) FOR THE TEST RUNNING SDMM49

REFERENCES

- [1] J. Hariyono, K. Jo, Pedestrian Action Recognition Using motion Type Classification, Proceedings of 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), 2015, pp.129-132.
- [2] R. Poppe, A Survey on Vision-Based Human Action Recognition, Image and Vision Computing 28, 2010, pp.976-990.
- [3] N. Sun, Y. Sakai, New Approaches to Human Gait Simulation Using Motion Sensors, 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, Mar. 2017