C-01

Surface Detection using Accelerometer Sensor Time Series Data in 4-legged Robot

Saengrungruangsri Sutthikorn†　　Atsushi Ueno†　　Shoji Tatsumi†

1. Introduction

Creating effective robot motion to maximize their performance is substantially challenging task. Since particular environments which potentially affect speed or stability of robots might be considered as one of crucial factors affecting the process of defining the best motions. Therefore, the calibration of individual specific environment is necessary in order to keep the robots operating at peak performance.

In recent years, the accelerometer-based gait or activity recognition has been studied extensively. Kwapisz et al. [2] developed an activity recognition system using phone-based accelerometer in Android-based smartphone to identify the physical activity which a user is performing. He & Jin [5] argued that FFT is too costly for real-time computation. Their approach is using AR model instead of time-domain features to represent the signals from the tri-axis accelerometer carried by user. Cho et al. [6] reported 93% overall accuracy on their proposed method using SVM on tri-axis accelerometer data to recognize user's activity.

A 4-legged robot Sony AIBO equipped with tri-axis accelerometer which can enhance its ability to recognize and interact with environment better. Vail et al. [1] proposed learning approaches that use data from AIBO's built-in accelerometer sensor to model the state of a robot or the state of its environment. As a result, the accuracy of overall performance of surface recognition can reach up to 84.9%. They created a decision tree using C4.5 algorithm and used 10-fold cross validation to quantify the accuracy of the final classifier over 6-features data which consists of variances in x, y, and z accelerations along with the (x, y), (x, z), (y, z) correlation coefficients over each data window.

In this study, we propose an approach of extracting pattern of time series gathered from AIBO's built-in accelerometer sensor which will be used for the surface detection process. In our first experiment, we reduce dimension of labeled-data by resampling data at several sample rates and reconstructed the phase space for each sample rate before clustering the data using k-means algorithm to compare error rate of each sample rate. Furthermore, the process of pattern extraction and surface detection are described in the reminder of this paper.

2. Data Dimension Reduction

2.1. AIBO

Sony AIBO ES7 which equipped 64-bit MIPS processor running at 576 MHz and 64MB of main memory inside was used in this study. The robot has total 20 degrees of freedom (DOF) consisting of three for head, one for mouth, three for each legs, one for each ear, two for tail. In addition to its 350,000 CMOS camera, AIBO also equip with Distance sensor (in its snout and on its chest), Head touch sensor, Back touch sensors, Chin sensor, Paw sensors, Vibration sensor, and tri-axis Accelerometer sensor. The accelerometer sensor operates at 125Hz and has value in the range of [-2,2] gravities or [-19.613300,19.613300] m/s^2 for each axis. x-axis depicts front-back direction (positive x value indicates robot's front size), y-axis represents left-right direction (positive y value indicates robot's left side), and z-axis represents up-down direction (positive z value indicates robot's up size). The sample accelerometer data is demonstrated in figure 1.

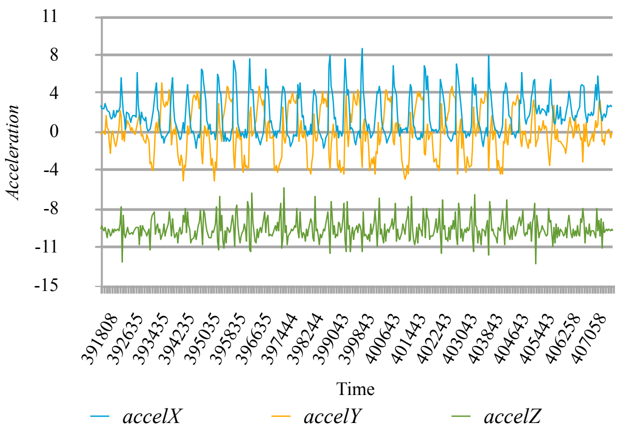

Figure 1. 15-second data acquired from tri-axis accelerometer while AIBO is walking on concrete.

2.2. Urbi Platfrom

In our experiment, we use Urbi Server 2.0 and UrbiLab for remoting and data acquisition.

Urbi is an open source cross-platform based on a parallel and event-driven script language, and on a distributed component architecture which is used to develop portable applications for robotics and artificial intelligence. The lack of consensus about standards of robotics result in greater susceptibility of the Urbi platform to the differences of Application Program Interfaces (APIs) among robots, components and so forth. The Urbi platform architecture is illustrated in figure 2.

2.3. Resampling Data

Chaotic data is observed often in experiments in many fields and traditional technique such as Fourier analysis are not effective when dealing with such chaotic signals. According to the studies conducted by Takens in 1981 and Sauer et al. in 1991, time delayed coordinates of the signal are used to construct a new set of coordinates. Data $x(t)$ where $t = 1 \cdots N$ in time domain is transformed into time delay domain $x(t+\tau)$.

AIBO's accelerometer sensor operates at 125Hz, in other words, 125 data points will be generated in 1 second on each axis (375 points/second in total). As a result of limited system resources, AIBO can not deal well with such huge amount of

† Graduate School of Engineering, Osaka City University

real-time data. We resample 15-second data while AIBO is walking on concrete at 8, 10, 12.5, 25, 40, 50Hz and perform time-delayed phase space reconstruction for each sample rate follow by comparing error rate in clustering data using k-means algorithms. Given k is a pattern size of each sample rate.

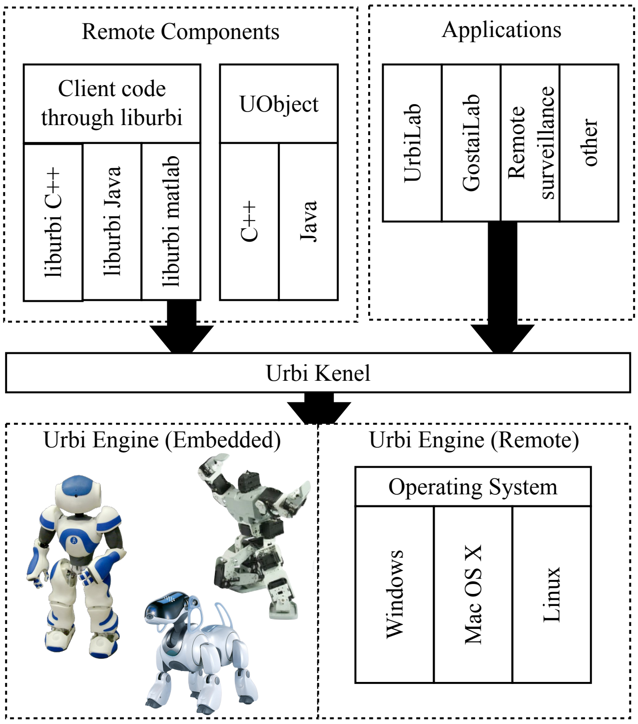

Figure 2. Urbi Platform Architecture

Since each time data set has to be stationary before performing phase space reconstruction, we trim off head and tail of data set (the data when AIBO is increasing speed and start walking, and the data when AIBO is decreasing speed and finally stop walking) and perform phase space reconstruction using time delay $\tau = 1$. As described in figure 3, the increase in frequency of sample rate causes an inadequate distribution of data in phase space.

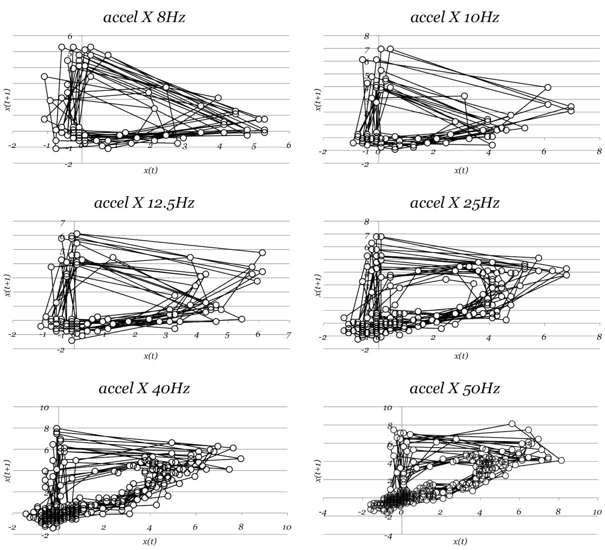

Figure 3. Phase space of each sample rate

Data is labeled as pattern point before being clustered using k-means. Figure 4 describes error rate of comparison labeled data with clustering result. The accuracy of clustering decreases continuously while sample rate increases. At 50Hz sample rate, error rate on x-, y- axis reach 90% that might interprets as practically unusable.

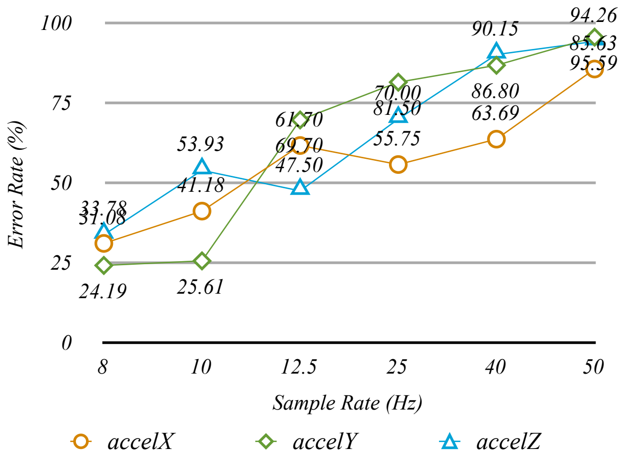

Figure 4. Error rate of comparing labeled class with k-means result of each sample rate

Considering both the accuracy and the cost of computation, we might, therefore, be reasonably conclude that 8Hz is the most suitable sample rate for using in our approach.

3. Surface Detection

3.1. Pattern Extraction

Before it is possible to enter into a detailed discussion of pattern extraction, we must try to clarify our principle concept of pattern. In this paper, we use the term "pattern" to refers to a pattern that occurs frequently in a data set, or "frequent pattern" in time series.

We gather 15-second tri-axis accelerometer sensor data at 8Hz while AIBO is walking on concrete and carpet. Each surface comprises of 5 sets of data for training and another 5 sets for testing. Each time series undergo preprocess to ensure that its mean value is constant over time, in other words, stationary time series. We then reconstruct phase space for each axis of data set and compute distance from each pattern point $P_1 \ldots P_n$ to all data points $d_2 \ldots d_N$ to construct distance matrix. Data in distance matrix are chunked into block of pattern-size in each column. The Euclidean distance is given as:

$$dist(P_i, d_j) = \sqrt{(x_d - x_p)^2 + (y_d - y_p)^2} \quad (1)$$

is used to compute distance between P_i to d_j, where P_i is point in pattern, and d_j is data point in time series. But, in order to diminish cost of computation, we derive distance equation as:

$$dist(P_i, d_j) = (x_d - x_p)^2 + (y_d - y_p)^2 \quad (2)$$

The distance matrix and the minimum values of each block is depicted in Table 1.

Since AIBO's gait is periodic movement, the period within walking can be called the gait cycle. The minimum value in each block indicates the point in one gait cycle that has the minimum distance to reference pattern point. By averages minimum values of each block, pattern points are extracted. The pattern point values and the slopes between 2 adjacent pattern points in pattern are used in our method to define the pattern. The slope can be computed by:

$$slope(d_i, d_{i+1}) = y_{i+1} - y_i \tag{3}$$

where d_i, d_j are adjacent points.

Table 1. Distance matrix of x-axis data. PAT refers to points in pattern. Highlighted cells indicated the minimum value of each block.

DATA/PAT	P_1	P_2	P_3	P_4
d_2	6.772433381	0	-	-
d_3	24.84173342	7.025356543	0	-
d_4	33.52504688	19.92397378	0.702545712	0
d_5	38.72381944	40.01651831	7.025386717	1.011651744
d_6	0.112405973	46.59221005	40.01651831	27.70804986
d_7	46.81701731	0.449623892	11.52159479	31.58606699
d_8	37.99319403	20.51406625	0.562029194	3.653204177
d_9	38.4990085	43.72590267	8.823878595	1.124057047
d_{10}	0.14050579	45.66488928	38.24617687	25.99390973
d_{11}	41.59023277	0.056202987	12.70186555	31.75470981
d_{12}	42.54560844	18.20977968	0.281015603	5.957505835
d_{13}	34.424294	43.52918987	9.357808308	0.702534984
d_{14}	17.56339136	36.22278956	10.48187608	1.264572224

We apply the extracted patterns on test data to perform a cross checking. The average accuracy in classifying data using extracted patterns was shown in table 2.

Table 2. Accuracy of classified test data using extracted patterns.

	x-axis	y-axis	z-axis
carpet	83.81%	82.85%	62.05%
concrete	85.09%	74.56%	57.17%

3.2. Surface Detection

In research on surface detection in robot, high accuracy method, often fails to grasp computation cost. Numerous attempts have been made offline on server to reach high precision. Conversely, our approach would rather focus on enabling computation to be performed on robot itself.

We use sliding window algorithm in pattern matching process, and define a window width equal to the size of pattern (4 on x-axis and z-axis, and 8 on y-axis). The window moves step by step for searching the patterns. Window will move the length, if the pattern is found, or, it will move just one point until the pattern is found. To perform pattern matching, we use both point's value and slope of 2 adjacent points. The value describe the similarity of data to each point in pattern, while slope describe direction of pattern. The surface is recognized only if detected pattern of 2 in 3 axises refers to the same surface.

In the experiment, we gather data while AIBO is walking from carpet to concrete, and from concrete to carpet. We then employe extracted patterns to detect surface at that time by simulate on matlab. Figure 5 shows the result of simulating surface detection using matlab.

We, then embed extracted patterns in Urbi's default walking script and revise the script for enabling AIBO to detect surface while it is walking. The sliding window algorithm is also employed in detecting surface on robot. In contrast with conducting surface detection on server which full time series data is already available, detecting surface on robot must collect data point by point on each axis, and then match observed data point to pattern points. In the experiment, we achieve average overall accuracy at 75%, but slow response is slightly detected. One possible reason is, the response might result from time-consuming during the computation of AIBO as the complete set of data is required to be filled in the window before defining the surface.

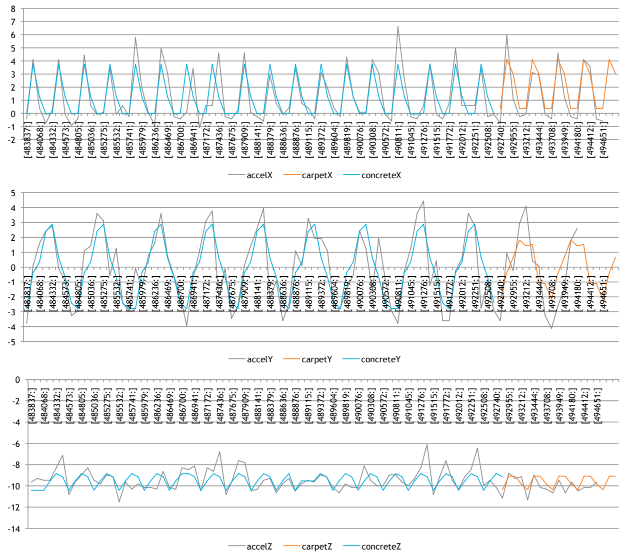

Figure 5. Detecting surface using data on each axis. The observed data, the pattern while walking on concrete, and the pattern obtained during walking on carpet are represented by gray, blue, and orange lines respectively.

4. Discussion

The overall accuracy of detection result obtained from the approach are approximately 76% for carpet, and 72% for concrete. Our method can not deal with certain non-linear variations in time dimension in time series, e.g. lag in time series and etc.

Dynamic Time Warping (DTW) is a method allows a computer to find an optimal match between two given sequences with certain restrictions. Comparing DTW method to the Euclidean Distance, the DTW is tedious method that consumes greater both time and resources for the computation.

Regards we use only 4 pattern points on x-axis and z-axis, and 8 pattern points on y-axis for pattern matching on 8Hz time series. It might be considered as a small pattern-size which can be treated effectively with the DTW technique. In future work, we will implement DTW technique for the process of pattern matching in surface detection, with the aimed of achieving higher accuracy.

Recently, the Urbi has been recognized as an universal platform for robot development. As we used the Urbi as development environment, therefore, the approach employed in this study can be widely applied for the further legged-robot projects.

5. Summary

We propose a low-computation cost method in surface detection using AIBO's built-in accelerometer sensor data. This approach differs significantly from the others that generally emphasize only the increase in accuracy. The results from the experiment show that the use of pattern extraction together with surface detection can reduce computation cost with acceptable levels of accuracy.

6. Reference

[1] Douglas Vail, Manuela Veloso., *Learning From Accelerometer Data On A Legged Robot*, In proceeding of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal 2004

[2] Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore., *Activity Recognition using Cell Phone Accelerometers*, In proceeding of the 4th International Workshop on Knowledge Discovery from Sensor Data (SensorKDD-2010), Washington, DC, USA 2010

[3] Michael Fitzgerald Nowlan., *Human Recognition via Gait Identification Using Accelerometer Gyro Force,* In proceeding of the 7th ACM Conference on Embedded Networked Sensor Systems, Berkeley, CA, USA 2009

[4] Jani Mäntyjärvi, Mikko Lindholm, Elena Vildjiounaite, Satu-Marja Mäkelä, Heikki Ailisto., *Identify users of portable devices from gait pattern with accelerometers*, In proceeding of International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, USA 2005

[5] Zhenyu He and Lianwen Jin., *Activity recognition from acceleration data using AR model representation and SVM*. In proceeding of International Conference on Machine Learning and Cybernetics, Hong Kong, China 2008.

[6] Yongwon Cho, Yunyoung Nam, Yoo-Joo Choi, and We-Duke Cho., *Smartbuckle: human activity recognition using a 3-axis accelerometer and a wearable camera*. In proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments, Breckenridge, CO, USA 2008.

[7] Sonia Chernova, Manuela Veloso., *An Evolutionary Approach To Gait Learning For Four-Legged Robots*. In proceedings of In Proceeding of International Conference on Intelligent Robots and Systems, Sendai, Miyagi, Japan 2004.

[8] Jean-Christophe Baillie., *URBI: Towards a Universal Robotic Low-Level Programming Language*. In proceeding of International Conference on Intelligent Robots and Systems, Alberta, Canada 2005

[9] Jean-Christophe Baillie, Akim Demaille, Quentin Hocquet, Matthieu Nottale, Samuel Tardieu., *The Urbi Universal Platform for Robotics*. In proceeding of International Conference on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS, Venice, Italy 2008

[10] Gostai., *The Urbi Software Development Kit 2.0*

[11] Gostai., *Uribi Doc for Aibo ERS2xx, ES7 and URBI 1.0 : Devices Document*

[12] David Filliat., *Introduction to liburbi-matlab*