Exploring Surface Detection for a Quadruped Robot in Households

 Csaba Kertész  
 Vincit Oy  
 Hermiankatu 3 A, 33720 Tampere, Finland  
 csaba.kertesz@ieee.org

Abstract—Surface recognition is essential for legged robots because they need to maintain their dynamic balance on a regular or uneven terrain. The accelerometer is a widely-used tool for this purpose, but the quadruped Sony AIBO does not have such a high-end sensor compared to the latest state-of-art developments. Past works focused on attaching replacement sensors to the robot dog as well as collecting many samples for machine learning methods although some studies did not address this issue at all. This paper focuses on improvements with sensor fusion of built-in sensors to recognize wider variety of surfaces and get similar or better accuracy than earlier experiments. The combined features are based on the accelerometer and force sensors in the leg joints to handle this problem. Evaluation suggests that this method can reduce the data collection time for training samples dramatically and it is suitable for practical applications.

Keywords—surface recognition; accelerometer; Oscillation Power; Sony AIBO

I. INTRODUCTION

Legged robots can be more effective on certain terrains than wheeled robots, but their gaits must be adapted to the underlying surface. This adjustment can be aided by providing information about the surface material or basic properties like rigidity. The state-of-art methods use camera, accelerometer and force sensors in the leg joints to handle this problem.

In [8], features were extracted from joint motor currents and ground contact force measurements to estimate different terrain shapes and surface features. The classification was done by a multiclass AdaBoost machine learning algorithm and the system performed well. Another terrain classification [22] used forces and torques measured on a six-legged walking robot and the system recognized five surfaces with a success rate of 76%.

Accelerometers measure linear accelerations in the x, y and z dimensions and sophisticated applications can determine other features from temporal signal oscillations. The popular Kalman filter was applied to a MEMS sensor by Qin et al [15] to detect the posture of a wheeled robot, while in a particular research [13], an extra Inertia Measurement Unit (IMU) device was mounted on AIBO to stabilize the head during locomotion. In [9], the directional bias was integrated from the x and y dimensions of an accelerometer. Sinha and Bajcsi [18] put an accelerometer to their robot’s foot and they were able to detect when the legs were slipped as a surface property.

Though the high-end IMUs on the market include an accelerometer and a gyroscope to combine their readings for more accurate inclination estimations, Sony AIBO is equipped with a simple accelerometer [19] which challenges the precise detection of body oscillations during locomotion. Contrary to previous solutions when four-legged robots had been modified with custom hardware, Vail and Veloso [21] implemented a C4.5 decision tree classification built on the variance of single and pair-joint variances of the three accelerometer dimensions (x, y, z, x-y, y-z, x-z) to detect surfaces with AIBO. Their approach did not require manual calibration, but a large sample database was collected and a longer off-line learning process was needed for good performance. Chen et al [5] developed an artificial vestibular system to estimate the live posture, slope and surface with the built-in accelerometer of AIBO. Their surface recognition was based on a new Oscillation Power signal extracted from the high-pass filtered accelerometer signal. Although the OP descriptor has been defined, its evaluation did not go into details.

Past works with other robots implemented the surface detection based on accelerometers, motor currents or ground contact forces while earlier studies with AIBO used only accelerometer. In this paper, two separate types of sensors are combined to increase robustness. Higher level statistics are obtained from OP of the x, y and z accelerometer dimensions and other features describe how much the paw sensors are activated by ground contact during locomotion. The built feature vectors were input to machine learning methods and a couple of classifiers were competed to select which one performs the best. Chen et al [5] have not tested any machine learning algorithms for their OP, Vail and Veloso [21] implemented C4.5 decision trees without deeper analysis of different methods. After the new method is evaluated against other approaches with legged and wheeled robots, a subchapter presents how the data collection time for training data can be reduced at the cost of a 5-7% performance drop.

II. EXPERIMENTAL SETUP

A Sony ERS-7 (Fig. 3) was used in the experiments, and since AIBO models have weak servomotors, trot or pace did not suit as locomotion gait because they put an unpleasant stress on the motors and the gears. A slower, singular crawl gait was developed based on a MoNet [24] walking style from the official Open-R SDK examples. Fig. 1 shows how the legs move during a walk period ($\phi$), where the black bars indicate the ground contact of the four legs which swing forward during the white periods.
Fig. 1: Singular crawl gait. The black area indicates if a leg is contacted with the ground while pushed backward, otherwise it swings forward in the air.

This crawl gait is quite stable because three legs have always ground contact, and hence, the robot can maintain balance easily compared to other gaits like trot.

AIBO is an indoor entertainment robot, therefore, it was run on five different types of surfaces commonly found in households. Wood flooring with lacquer coating (Fig. 2.a/Fig. 2.b) and vinyl flooring were the most rigid ones. Their opposites were a 13 mm thick carpet (Fig. 2.b) and foam mats typical for homes with small kids. The fifth surface was a thin carpet (2 mm) which had similar properties than the field used in RoboCup competitions.

There is an accelerometer (Fig. 4) in the torso with a 120 Hz sampling rate though other robots in some recent works used high-end accelerometers with a sampling rate of 44.1 kHz [3] and 4 kHz [7]. Each leg of the dog has a paw sensor with a 10 Hz sampling rate and the plastic paws of AIBO can not maintain their position on a slippery ground. As a resolution, a set of dog socks (size L) were drawn on the legs (Fig. 3). Their anti-slip strips were pulled back on the front paws and forth on the back paws, following the main ground contact areas during locomotion.

A behavior-based motion control of the AiBO+ open-source project1 played back the MoNet key frames for the leg joints and interpolated all intermediate positions. The robot walked on five surfaces and the sensor data were transferred to a laptop via wireless network. In one test run, the dog was started in a lying position and after standing up, it walked at a constant speed for 30 seconds, during which it completed 12 to 14 walk periods depending on its speed. 50 samples were gathered for each surface type (|$S_{\phi,\text{wood}}$| = |$S_{\phi,\text{field}}$| = |$S_{\phi,\text{carpet}}$| = |$S_{\phi,\text{foammat}}$| = |$S_{\phi,\text{vinyl}}$| = 50) thus $S_{\phi}$ has 250 samples in total. The sample sets have 375 minutes data which is a substantial increase to what previous works have gathered for surface detection: 15 minutes with AIBO [21], about 12 minutes with a leg prototype [8], about 60 minutes with a cardboard box [7] and 170 minutes with a wheeled outdoor robot [23].

A two walk periods long temporal window is chosen to extract features from the sensor data:

$$W_t = [t, ... , t + 2 \phi],$$

where the acquired time series length vary according to the function of the respective walk period. For example, $W$ is 4200 ms long when $\phi = 2100$, but $W = 4800$ ms for $\phi = 2400$. This duration provides a good basis to build feature vectors for machine learning methods (see Chapter III), with a compromise between a longer window with slower response time and a shorter one with noisier results. When the robot walks from one surface to another, the transition time takes about 2.5-3 walk periods, thus a long window may fail to detect these changes properly.

Since the accelerometer and the paw sensor sampling rates are different, the time series from these sensors in a $W$ window were defined as follows:

\[ S_{\phi} = \{S_{\phi,\text{wood}} , S_{\phi,\text{field}}, S_{\phi,\text{carpet}}, S_{\phi,\text{foammat}}, S_{\phi,\text{vinyl}} \}, \]  

1 The project web address: http://aiboplus.sf.net
where \( A_x \) is the x dimension of the accelerometer, \( A_y \) is y and \( A_z \) is z respectively. Paw sensors are denoted by \( P_{LF} \) (left-fore), \( P_{RH} \) (left-hind), \( P_{RF} \) (right-fore) and \( P_{BH} \) (right-hind) in Equations 6-9.

### III. Feature Vector

If labeled training samples are available to use supervised learning, surface detection is a classification problem for the robot. The raw data are never fed to a classifier, but higher-level features are extracted to represent the sensor information in a compressed form (feature vector). In this paper, the feature vector implements sensor fusion because it is composed by statistical descriptors of the accelerometer and the paw sensors data:

\[
FV(W) = [f_{v0}, f_{v1}, ..., f_{v9}],
\]

where the feature vector is built upon data of a \( W \) window and there are 10 numbers \( (f_{v0}, f_{v9}) \) in one vector. The first six features are related to the accelerometer and four to the paw sensors. Comparing to previous works, Vail and Veloso [21] used 6 features derived from the accelerometer data, Weiss et al [23] generated 128 features from fast Fourier transform of the data of z-axis and Hoepflinger et al [8] defined 24 features from motor currents and ground contact forces.

#### A. Accelerometer Sensor Based Features

When a robot walks on a rigid surface, it produces vertical body oscillations while soft surfaces absorb these anomalies. Based on this assumption, Chen et al [5] applied a high-pass filter to the time series of the accelerometer and unnecessary low harmonics were eliminated to get an estimation of these oscillations. Their Oscillation Power feature can be calculated here as follows:

\[
OP_x = HF(A_x)^2 / 10^3,
\]
\[
OP_y = HF(A_y)^2 / 10^3,
\]
\[
OP_z = HF(A_z)^2 / 10^3,
\]

where \( HF \) is a 6-order Chebyshev I high-pass filter with 12.5 Hz cut-off frequency and 0.0873 ripple dB.

In [5], \( OP_z \) showed the most significant ability to detect different surfaces, and similarly in [23], the z dimension of the accelerometer data were collected to recognize vibrations. Fig. 5 shows \( OP_z \) of the third-fourth walk periods \( (W(2g)) \) generated from 50 runs per surface. The medians show a decreasing tendency towards the softer surfaces except for the foam mat which is on the same level as wood. This phenomena may be caused by the elasticity of the foam which makes it work as a compression spring: it stores some energy when the leg is down on the ground, and which gets released when the leg is lifted to swing forward. In general, the overlaps of the interquartile ranges (IQRs) among surfaces cannot guarantee a good classifier performance based on \( OP_z \).

Due to the variance of \( OP \), statistics were derived for \( FV \). Similar to the usage of joint relationships between accelerometer dimensions in [21], here, the rates of medians, means and the joint interquartile range quotients of \( OPs \) were chosen as measures. The first describes the span of the body oscillations while the latter can indicate their variances. Six features are denoted as:

\[
f_{v0} = \text{mean}(OP_x) / \text{median}(OP_x),
\]
\[
f_{v1} = \text{mean}(OP_y) / \text{median}(OP_y),
\]
\[
f_{v2} = \text{mean}(OP_z) / \text{median}(OP_z),
\]
\[
f_{v3} = \text{iqr}(OP_x) / \text{iqr}(OP_y),
\]
\[
f_{v4} = \text{iqr}(OP_z) / \text{iqr}(OP_y),
\]
\[
f_{v5} = \text{iqr}(OP_z) / \text{iqr}(OP_y),
\]

where \( \text{mean} \) is an arithmetic mean statistic and \( \text{iqr} \) denotes the size of an interquartile range. The \( f_{v0}-f_{v2} \) components are computed from the distinct accelerometer dimensions, but \( f_{v3}-f_{v5} \) represent the pair-joint correlation measure between these dimensions \( (x-y, y-z, z-x) \).

#### B. Paw Sensors Based Features

AIBO cannot measure precise ground contact forces, unlike in [8], since its paw sensors are simply two-state buttons \((1 - \text{pressed}, 0 - \text{not pressed}) \). These buttons get pressed more when the robot is walking on a hard surface than they would on carpet, therefore, it is measured how many times these sensors are put into a pressed state during locomotion:

\[
f_v = \sum_{i=6}^{9} \left( P_{leg} / |P_{leg}| \right), \quad \text{leg} \in \{LF, LH, RF, RH\},
\]

where the press counts are added up and divided by the number of sensor samples to get the duration of the pressed state in the percentage of a walk period.

#### C. Feature Scaling

Although the standardization and scaling to feature vector
length were examined by the author as a preprocessing step before the classifier training, it did not improve the results. It was enough to normalize the feature vector elements to similar magnitudes to avoid unbalanced training. Some features are scaled as follows:

$$f_{v_i} = f_{v_i} \times 10^{u_i}, \quad i \in [0, ..., 9],$$  \hspace{1cm} (22)$$

where the first three accelerometer features should not be normalized ($M_{a,0} = 0$) and the second three should by 10 ($M_{a,3} = 1$). The percents of the paw sensor-based features must be multiplied by 10 ($M_{r,0} = 1$).

IV. SURFACE RECOGNITION

A. Classifier Selection

The k-fold cross-validation is a standard comparison of how well the classifiers can generalize their hypothesis for a given problem. While they may yield comparable results when large amount of training data are available, both the sample and feature count are low in this paper. To do this inference, the averaged results of the iterated k-fold cross-validations were calculated because they provide a nearly unbiased estimate of the algorithm performances and they reduce the variances of the estimators and outperform the 632+ bootstrap method [10].

K-fold cross-validation is applied to a classifier over $S_s$ in the following way. Extract the feature vectors from all samples of $S_s$ in a $W_t$ time window and put them in a set $F_{v_s}$. Randomize the order in $F_{v_s}$ and split it into k equal parts (folds). Train a new classifier with k-1 folds (training set) and check the performance on the k-th fold (testing set). Repeat this step until each fold is evaluated against the other k-1 folds and calculate the arithmetic mean of these performances ($\bar{C}_{v_s}$). A practical question is how many iterations are enough to converge towards an unbiased result. 20 iterations were found suitable to give a good guess, 100 iterations were accurate to one decimal and over 2000 iterations to two decimals.

There is no established way to choose the k parameter in the k-fold cross-validation. The author run repeated 5-, 6-, 8- and 10-fold (2000 times) and Monte-Carlo cross-validations (20000 times) with a training set size of 80-90% of the sample set. Results showed nearly constant accuracy across all classifiers examined in this chapter if the training set sizes were equal for both k-fold and Monte-Carlo. Eventually, the common 10-fold cross-validation has been selected.

Several classifiers were challenged with the feature vectors generated from the third-forth walk periods to find the best performing one. The initial body oscillations (first-second walk periods) have an undesired effect on the extracted features as noise, therefore, those were omitted. In [22], the duration of the first step was excluded with similar purpose for their hexapod.

The machine learning algorithms were used mostly from the Dlib [11] and OpenCV [12] libraries. Extensive learning parameter searches were performed to discover the sensibility of each classifier and broad value intervals showed similar performances therefore the learning parameters were picked randomly. Albeit the averaged k-fold cross-validations give an unbiased estimate about classifier accuracy, the distributions of the iterated k-fold results may be used to compare the variance of the classifier performance. Fig. 6 shows 2000 times repeated 10-fold cross-validations ($\bar{C}_{v_{2000}}$) for each method where a higher median represents better accuracy and a narrower interquartile range (IQR) stands for smaller variance. Naive Bayes (NB) [6] was the top performer (median: 93%) with the uppermost and narrowest IQR. C-SVM (support vector machines) [4] applied c-support vector classification ($C = 0.001$) with linear kernel and achieved a bit lower results (91.57%) than decision trees (DT) [2] (89.56%). v-SVM follows with 89%, v-support vector classification ($v = 0.0001$, $\gamma = 0.00001$) and radial basis function kernel which is a moderate result like maximum entropy (ME) [14] (L2 = 0.005) with 89.5% and kernel ridge regression [17] ($\lambda = 1.5$, linear kernel) with 83.43%. Although the relevance vector machines [20] are related to SVMs, their median is 73.1% the worst with RBF kernel ($\epsilon = 0.00001$, $\gamma = 0.00002$), along with k-nearest neighbors [1] (78.14%). Similar tendencies were observed in the classification performance for other walk periods. Naive Bayes classifier was selected for further experiments because it provided the most accurate results with low variance.

B. Performance Evaluation

The last subchapter explored the classifier performance based on the third-forth walk periods and naive Bayes was the best. This approach showed a preliminary suggestion about the classifier selection, but the previous works measured the recognition accuracy on longer runs of the robot [21, 22, 23]. In this subchapter, the time window ($W_t$) is shifted after the forth walk period until 30 seconds in order to determinate how naive Bayes performs on more sensor data:

$$CCV_{\phi}(t) = \frac{\sum_{i=0}^{2000} \bar{C}_{\phi,i}}{2000},$$  \hspace{1cm} (23)$$

where 10-fold cross-validation is executed on the extracted feature vectors 2000 times after each movement of the window ($W_t$) and $CCV_{\phi}(t)$ contains the means of these repeated cross-validations for a walk period. In Fig. 7, $CCV_{\phi,1800}$ represents $\phi = 1800$, $CCV_{\phi,2100}$ is for $\phi = 2100$ and $CCV_{\phi,2400}$ for $\phi = 2400$ while a $CCV_{\text{mixed}}$ was generated from mixed samples of all walk periods in an iteration. $CCV_{\phi,1800}$ has the best results with an overall arithmetic mean $CCV_{\phi,2100} = 91.12%$ while $CCV_{\phi,2400}$ has 89.61%, $CCV_{\phi,1800}$ has 85.65% and $CCV_{\text{mixed}}$ has 86.59%.
Every function is in an interval $[80; 95]$ which suggests that this new surface classification can yield good performance regardless of the locomotion speed. These results are improvements compared to the 84.9% overall classification rate in [21], in which an ERS-210 robot dog was run with one walk speed on three different types of surface (cement, field, carpet), but the ERS-7 in this study was run on five types. Table 1 shows the confusion matrix for $CCV_{2100}$ and three surface types (wood, field, carpet) have good accuracy over 90% with low misclassification rates; carpet has the best score with 96.52%. Vinyl flooring and foam mat were mutually misclassified (12.15%, 14.34%) to certain extent. A higher level of misclassification had been assumed between wood-vinyl and field-foam mat pairs because of their similar rigidity, but the experiments did not confirm these expectations (Table 1).

In [23], 3-class and 7-class problems were examined using more classifiers. After they evaluated a classifier on the sample set of each speed, the worst performance was similar to the performance of a final mixed sample set. The same phenomena can be seen in this paper, as $CCV_{mixed}$ is a bit higher than $CCV_{1800}$ and its curve largely follows $CCV_{1800}$. The result of their 3-class experiment with significantly different surfaces were around 90-95% (SVM) and similar accuracy can be expected here for wood, field and carpet surfaces based on Table 1, but Brooks and Iagnemma [3] achieved lower performance (85.3%) despite their high-end accelerometer (see Chapter II).

Weiss’ 7-class experiment involved only six surface types because they included samples when the robot was stopped as a “surface class”. Their best performance with SVM was 77.6% over mixed samples of six classes and this study achieved 9% higher (86.59%) although with one surface less. A six-legged robot [22] recognized five surfaces on one walk speed with a 76% success rate and a leg design for quadrupedal robot [8] achieved 73.32% with four surfaces thus this novel method with (85.65%, 89.61% and 91.12%) are notable improvements over the earlier experiments using only one speed. Shadukhan [16] found rising accuracy when the speed was increased, however, the results here do not confirm the relationship between the walk period and the performance, similar to [23].

C. Performance Evaluation with Small Training Set

Past works focused on collecting a massive amount of data and evaluating classifiers on the whole set as the previous subchapter presented. This paper goes one step forward to explore the surface recognition with minimal training data. Instead of a temporal, repeated 10-fold cross-validation (Chapter IVB), all training vectors are extracted from the third-fourth walk periods and evaluations are done on the later sensor data of the testing samples. As Chapter IV.A pointed out, repeated k-fold and Monte-Carlo cross-validation showed almost identical results for this surface classification problem. Based on this, a modified Monte-Carlo cross-validation was defined for a walk period ($\phi$):

1. Randomize the order in $S$ and split it into training and testing set (90%/10%). (Note that the training/testing set sizes are equals to 10-fold cross-validation splits.)
2. Extract the training vectors from the third-fourth walk periods ($FV(W(2\phi))$).
3. Extract testing feature vectors from the testing set based on sliding windows ($W(2\phi), ..., W(30 \text{ sec}-2\phi)$).
4. Evaluate the performance between $r \in [\phi, ..., 30 \text{ sec}]$.
5. Repeat the steps 1-4 20000 times and average the results in time points $t \in [4\phi, ..., 30 \text{ sec}]$. An $MC_s(t)$ function contains the final results.

Fig. 8 shows $MC_{1800}$, $MC_{2100}$ and $MC_{2400}$. The overall means are $MC_{1800} = 78.46\%$, $MC_{2100} = 86.52\%$ and $MC_{2400} = 84.33\%$. Although the results are lower by 5-7% compared to Chapter IV.B, but all curves are in a value interval [75; 90]. Considering the training was done on the data of the third-fourth walk periods, the performance is still similar or better compared to earlier studies. This finding makes it possible to create surface classifiers by collecting training data with short runs of the robot on a given surface and the entire model building process can be reduced considerably.

| TABLE I. CONFUSION MATRIX OF $CCV_{1800}$. THE ROWS REPRESENT THE REAL SURFACES AND THE COLUMNS HOW THEY WERE CLASSIFIED. |
|-----------------|-----|-----|-----|-----|-----|
|                | W   | F   | C   | FM  | V   |
| W   | 93.30 | 6.41 | 0.42 | 0.02 | 0   |
| F   | 2.95  | 95.18 | 1.40 | 0.54 | 0   |
| C   | 0.64  | 1.70  | 96.52 | 0    | 0   |
| FM  | 0.11  | 3.42  | 0    | 84.29 | 12.15 |
| V   | 0     | 0.29  | 0    | 14.34 | 85.41 |

Fig. 7: Repeated 10-fold cross-validations with naive Bayes in 30 seconds. The figure shows the results for walk periods $\phi = 1800, 2100, 2400$ and mixed samples sets.
Fig. 8. Continuous Monte-Carlo cross-validation with naive Bayes until 30 seconds. The figure shows the results for walk periods $\phi = 1800, 2100$ and 2400.

V. CONCLUSIONS

Unlike earlier researches which attached an extra sensor to AIBO [13] or collected bigger sample database to overcome its noisy accelerometer [21], this paper implemented sensor fusion to get more reliable surface recognition with the built-in sensors. Several classifiers were examined and naive Bayes was found the best for building terrain models. The developed classifier distinguished five different surfaces types commonly found in households with similar or better accuracy compared to the past literature and the recorded sample database contain data for an exceptionally long duration (more than 6 hours) to verify the findings of this new method. The paper also explored the ways of practical implementation with using the data of the third-forth walk periods for training exclusively to minimize sample collection time.

Future work can consider the fusion of more sensors to get increased reliability and include other surface types in the experiments. Using texture features of the ground would aid recognition, however, the camera of AIBO works well only in very bright environments. An interesting aspect would be to explore the transition detection between underlying surfaces or do a deeper analysis while walk speed is changed dynamically.

ACKNOWLEDGMENT

Thanks for the support from my company and the patience of my family while the sample sets were collected at home.

REFERENCES