Classification of Running Surface on an Embedded System - a Digital Sports Example Application

B. Eskofier, J. Hornegger,* M. Oleson, I. Munson, B. Krabbe, C. DiBenedetto†

eskofier@cs.fau.de

Abstract

This paper presents methods for collecting and analyzing biomechanical data of runners in order to classify the prevailing running surface situation. For this purpose, we consider a particular embedded application example from the field of digital sports: A novel running shoe that is capable of sensing run-specific parameters and adapting the cushioning setting accordingly. Shoe heel compression was recorded continuously during runs on a predefined test course. Each single step of the 24 participating athletes was labeled according to the surface situation. In total, data from 22910 single steps was collected for analysis. The data was analyzed using step features to decide what surface the runner was on with an accuracy of more than 80% across multiple study participants. The results show that for most participating runners, three step features are best suited for surface classification. This information is used in the current version of the aforementioned running shoe to adapt the shoe setting correctly.

1 Introduction

The ability to perform accurate classification in real time is a key factor for many applications. This is not only true when computationally powerful hardware is used. It is most often crucial in the restricted hardware environment of the power-efficient, highly mobile microprocessors used in embedded systems. The most important question is which of the complex algorithms known in pattern recognition can be used and implemented in the context of the restricted memory capacity and computational power of the employed microprocessors. Special considerations have to be made in order to adapt those algorithms to the specific hardware and classification task at hand. A lot of areas of engineering can benefit from the possibility of accurate classification in this restricted environment. Examples include, but are not limited to, automotive solutions, communications, industrial automation, speech recognition and medical care.

For this presentation, we focus on the application of these concepts on the adidas_1 running shoe, which is the first shoe that features an embedded system. This shoe is built to adapt to various running conditions like the prevailing surface situation. A precise classification of these conditions is of course mandatory to guarantee this functionality. To facilitate this, the step signal of the runner is continually measured and processed by the embedded microcontroller. A detailed description of the adidas_1, its functionality and embedded system hardware can be found in sections 2 and in DiBenedetto et al. (2004). We will describe the analysis methods that lead to accurate, real-time surface classification. To our knowledge, we are the first group researching step signal classification on an embedded system. The presented example for a classification system has recently been implemented in the current version of the adidas_1 running shoe. It is significantly contributing to the shoe's functionality and thereby offering runners an ideal adaptation during each phase of their run.

2 Materials and Methods

2.1 The adidas_1 Running Shoe

The adidas_1 is a running shoe that possesses a built-in 8-bit microcontroller, a sensor for heel compression measurement and a motor for cushioning adaptation. This shoe is designed for avid runners, and is constantly adjusting itself to the running situation. In this presentation, we will focus on the classification of the surface that the athlete is running on. The general demand to establish constant cushioning when a

*University of Erlangen-Nuremberg, Department of Computer Science, Chair of Pattern Recognition, Martensstrasse 3, 91058 Erlangen, Germany.

†adidas AG, adidas innovation team, 5055 N Greeley Ave, Portland, OR 97217, USA.

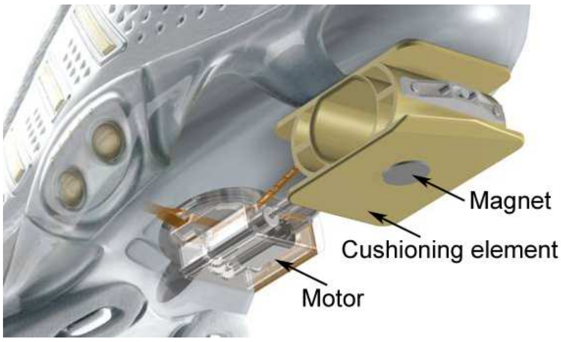

Figure 1: The adidas_1 shoe.

change of running surface takes place and all other running conditions remain constant is to have

- a soft shoe on hard surfaces (asphalt, concrete)
- a hard shoe on soft surfaces (grass, trail).

The automatic adaptation ideally takes into account the athlete's weight, speed, fatigue level and furthermore the current surface condition, elevation profile and shoe condition.

To facilitate this adaptation, the shoe features a cushioning element (Fig. 1), whose ability to give way in vertical direction can be regulated by a motor-driven cable system. The regulating cable is running from the motor through the middle of the cushioning element to its opposite end and is fixated there. The motor shown in Fig. 1 can adjust the attenuation setting by turning a screw which lengthens or shortens the cable. When the cable is shortened, the cushioning element is tensed and compresses very little when external forces are applied. When the cable is longer, it allows the cushioning element to compress further by giving it more room to expand in the x-axis direction (forward-backward direction), effectively making the shoe softer. For more details on the shoe design the reader is referred to DiBenedetto et al. (2004).

Compression measurement is made by a hall sensor that is mounted at the top of the cushioning element. It detects the magnetic field strength induced by a small magnet, see Fig. 1, and can be sampled with a rate f_s of up to 1 kHz. The sensor-magnet distance d_m is then computed from the magnetic field strength with an accuracy of ± 0.1 mm. A decision whether the attenuation of the shoe has to be adapted is made based on the measured sensor data, see section 2.2.

The sensor-magnet distance is sampled by the built-in microprocessor that is mounted on a flexible circuit board on the motor element. Currently, a Cypress Semiconductor Corporation controller CY8C21634 is used. It possesses a clock speed f_{clock} of up to 24 MHz, 512 Bytes of SRAM and 8 kByte flash program store.

2.2 Sensor Data

In order to get the data needed for the analysis, there is a special prototype system equipped with an interface for data collection. The data from the magnet sensor is stored with a 256 kByte EEPROM array and can be evaluated offline in a later stage. An example running signal is depicted in Fig. 2 (bottom) with the sensor-magnet distance d_m plotted against time t. During the time where the shoe is in the air, the measured signal consists mainly of noise. In contrast, the heel compression and decompression phases of the runner's steps can clearly be distinguished. This measured signal is the basis for the surface classification experiments in section 3.

2.3 Preprocessing and Labeling

To extract the individual steps that need to be classified, we first establish a baseline value $d_{m,base}$. This value corresponds to the sensor-magnet distance when the shoe is in the air between steps. It can be reasonably assumed that it is the most frequently occurring value in the data. Next, all sample values that belong to a compressed state are detected. Initial experiments substantiated that compressed states occur when the sample values are below a distance threshold $d_{m,thres} = d_{m,base} - 1.5 * \sigma_{data}$, where σ_{data} is the overall standard deviation of a dataset.

We define the start and end of the compression phase as those points in the compression states where the distance from $d_{m,base}$ drops below three sample units, which corresponds to 0.7 mm. By using this approach, all steps could be identified in the datasets. This was confirmed by manually extracting 449 steps in 6 datasets and comparing the manual and automatic approaches. The results were identical.

In order to learn the necessary parameters for class separation, we implemented a graphical user interface for data labeling. Each step is assigned to one of the classes manually. The labels are then used for the training of the classification system.

2.4 Feature Computation and Reduction

In our case the selected features contain the information of one step. For each step, we extracted 19 features from the heel compression signal denoted by $F1 \ldots F19$. Fig. 2 shows $F1 \ldots F10$. Features $F11 \ldots F19$ are all measuring the standard deviation (SD) of different attributes (see Tab. 1), SD is computed as follows:

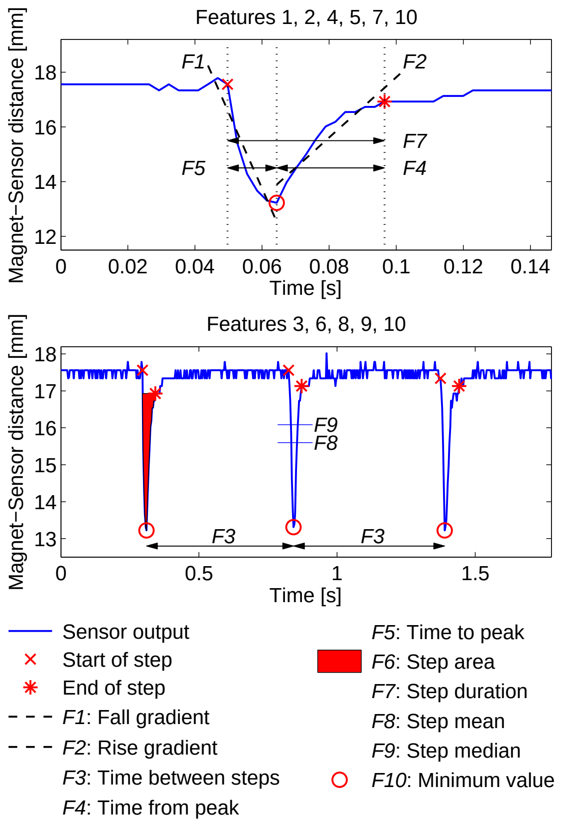

Figure 2: Step signal features 1 to 10.

$$\mathrm{SD}(\boldsymbol{c}) = \sigma_N = \left(\frac{1}{N-1} \sum_{k=1}^{N} (c_k - \bar{c})^2 \right)^{\frac{1}{2}} \tag{1}$$

where $\boldsymbol{c}$ is a vector containing the attribute data and varies depending on the attribute under investigation and N is the number of samples, i.e. the number of elements in $\boldsymbol{c}$. $\bar{c}$ is the average value of all the elements of $\boldsymbol{c}$. More specifically, for $F11$, $\boldsymbol{c}$ contains all the heel compression distance measurements d_m collected during a single step event (Fig. 2 top). For $F12 \ldots F19$, $\boldsymbol{c}$ contains the most recent 16 values of the corresponding feature. For example, for $F12$, each element of $\boldsymbol{c}$ is the minimum d_m that occurred within a single step, i.e. $F10$ and $\boldsymbol{c}$ has a total of 16 elements, one min. d_m value for each of the 16 most recent step events (see Fig. 2 bottom). $F13 \ldots F19$ have similar $\boldsymbol{c}$ values. For more details see Tab. 1.

The obvious redundancy contained in the extracted

Table 1: Step signal features 11 to 19.

Feature	Feature Description
$F11$	SD - values contained in one step
$F12$	SD - step minima ($F10$)
$F13$	SD - step means ($F8$)
$F14$	SD - step standard deviation ($F11$)
$F15$	SD - step duration ($F7$)
$F16$	SD - step area ($F6$)
$F17$	SD - time between steps ($F3$)
$F18$	SD - time to peak ($F5$)
$F19$	SD - time from peak ($F4$)

features is volitional. It was a goal from the start to use only a subset of the given features to reduce complexity further, thereby using only features with small or no mutual dependence. Fur this reason, we implemented a beam search as proposed by Bisiani (2004). The results of the beam search will be shown in the next section.

2.5 Classifier Selection

For our intended goal of embedded system classification we focused on classifiers that could be implemented computationally efficient. Our choices included

- Neural Networks (NN)
- Support Vector Machines (SVM)
- Linear Discriminant Analysis (LDA).

In order to test these and other classifiers we used the WEKA toolbox, see Witten and Eibe (2005). This toolbox allowed us to compare a lot of different approaches on powerful PC hardware in order to identify the algorithm that is best suited for the microcontroller implementation. Our experiments (see section 3) proved that in our case LDA classification yielded comparable classification rates to other, more complex approaches. We therefore decided to train a computationally cheap linear polynomial classifier using LDA. The theory behind this and other approaches can be found for example in Duda et al. (2000).

3 Experiments

3.1 Collected Data

In order to get a sufficient random sample for the subsequent classification experiments, a test course was selected where the desired surface conditions were

Figure 3: Aerial view of the test course.

present. The test course is located on the campus of the Faculty of Engineering of the University Erlangen-Nuremberg. It is depicted in Fig. 3. All runners were asked to run 6 sections of about 150 meters:

- two runs on soft surface (grass) with constant speed
- two runs on hard surface (asphalt) with constant speed
- one run on changing surface, starting on grass, then switching to asphalt, and finally running on grass again, all with constant speed
- one run on hard surface with a change in running speed (constant speed, then acceleration to a fast jog after the first half of the distance).

Each participant was asked to run normally with a comfortable but constant speed for the first 5 sections. Shoe setting, time information and an athlete profile (weight, height, training frequency) was noted for every runner. In addition to the shoe signal, a Polar RS800 system with foot pod was used to get speed and step frequency information.

24 test runners participated in this data collection. Shoes with sizes 7, 9 and 11 were used for those experiments. A total of 106 datasets with different shoe cushioning settings was collected for the subsequent experiments. Tab. 2 shows the number of steps for each of the 24 test runners that were used for the classification experiments. They amount to a total of 22910 single steps with a fraction of 50.6% on soft surface. The data was labeled as belonging to soft or hard surface using the GUI described in section 2.3.

3.2 Feature Selection

The results of the feature selection algorithm described in section 2.4 are given in Tab. 3 (see Tab. 1 for details on the features). For this evaluation, we used the fact that the classification of single steps can be improved when additionally taking a context of preceding steps into account. In this case, a context of three steps was used by casting a majority vote over the single decisions. In the implementation for the final product solution, a longer context can be used, which leads to even better classification results (see subsection 3.4). We finally selected the feature triple 1, 12 and 17 for the implementation on the microcontroller for two reasons. First, with the three-feature implementation we used 98% of the available program memory of the the CY8C21634 microcontroller. Implementation of a fourth feature would not have been feasible with the selected processor. The second reason for the implementation decision was that we could show that even with calculating the features and classification decision, we could still sample with maximum sample rate and therefore meet the real time computation criterion.

The confusion matrix for the selected feature combination is given in Tab. 4. Sensitivity is 77.7% and specificity is 73.6%. This result shows that no class is significantly favored over the other.

Table 2: The 24 runners with number of valid steps. Shoe size of each participant is given in brackets.

AB	(7)	1307	HH	(9)	936	MW	(11)	1165
AC	(9)	1152	JM	(11)	541	RB	(11)	670
AM	(11)	1338	JP	(9)	1013	RS	(11)	384
BD	(11)	781	KH	(11)	903	SK	(7)	1273
BE	(11)	1206	KR	(9)	1326	SW	(9)	1240
CD	(11)	911	MA	(11)	898	TS	(11)	612
DE	(11)	1121	MP	(9)	914	TT	(11)	791
EK	(7)	961	MS	(9)	627	VD	(11)	840

Table 3: Results for the first 5 iterations of the feature selection algorithm.

Selected features	Classification rate
1,12	75.4%
1,12,17	76.3%
1,2,14,17	76.9%
1,2,5,14,17	77.0%
1,2,7,12,13,17	76.9%

3.3 Classifiers

With the described three-feature subset we additionally performed experiments using other classifiers to decide whether other approaches could enhance our

Table 4: Confusion matrix for features 1, 12 and 17.

	Class soft	Class hard
Classified as soft	9000	2993
Classified as hard	2583	8334

Table 5: Results for different classifiers. Context of three steps was used.

Classifier	Classification rate
Nearest Neighbor, k=1	71.2%
Nearest Neighbor, k=3	73.1%
Nearest Neighbor, k=5	75.3%
Support Vector Machine	76.1%
LDA	76.3%
Neural Network	76.5%

results. For cross-validation, 24 subsets were used, each consisting of the samples of one individual runner. The results of these experiments are presented in Tab. 5. It can be seen that only the Neural Network with 1 hidden layer and 5 hidden units slightly outperforms the Linear Discriminant analysis. However, the gain in classification rate is not significant. Moreover, the complexity of the classifier, which has to compute more multiplications and has to evaluate the sigmoid function, inhibits its implementation on the embedded system.

3.4 Final Evaluation on the Microcontroller

It was important to implement our classification algorithm on the microcontroller that is employed in the product to verify our results. Longer contexts of 16 steps were used for the implementation. Tab. 6 shows the results of these experiments. Classification rates of more than 80% could be achieved.

Table 6: Datasets used for the evaluation on the microcontroller.

Dataset description	Steps	Hard	Acc.
Park (grass, concrete)	3480	61.5%	82.8%
Only asphalt surface	995	100%	92.0%
Forest soil, no incline	4438	0%	90.8%
Forest soil and asphalt, running up/downhill	4448	65.9%	80.3%

4 Summary

For the realization of accurate surface classification using sensor output from the adidas_1, data was collected from 24 test runners on hard and soft surface. This data was labeled, and 19 features were extracted which were chosen because they consistently represent the step information. A classification system using linear discriminant analysis was then proposed. Using the classification rate as a criterion, a subset of three features was found that is suited to be implemented on the embedded system that is integrated in the running shoe. The described classifier has been found to be more than 80% correct, and has been implemented in the current version of the adidas_1 running shoe.

5 Future Work

First results indicate that other important conditions can be classified using the shoe signal. One example includes the state of fatigue of a runner. An adaptation of the shoe hardness setting to a fatigued condition is definitely imaginable. Additionally, we will analyze the effect of elevation profile and speed changes in order to be able to classify these parameters, too.

References

Bisiani, R., 1987. *Beam Search.* In Shapiro, S.C., ed., Encyclopedia of Artificial Intelligence, Wiley-Interscience, New York, NY, USA, 55-58.

DiBenedetto, C., Oleson, M., Roth, C. & Thompson, M., 2004. *Intelligent footwear systems.* United States Patent US 20040177531, European Patent EP 1582108.

Duda, R., Hart, P. & Stork D., 2000. *Pattern Classification.* 2nd edition, Wiley-Interscience, New York, NY, USA.

Spearman, C.,1904. *The Proof and Measurement of Association between Two Things.* Am. J. Psychol. 15(1), 72-101.

Witten, I.H. & Eibe, F., 2005. *Data Mining: Practical machine learning tools and techniques.* 2nd Edition, Morgan Kaufmann Publishers, San Francisco, CA, USA.