In Proceedings of the 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007), pp. 59-64.

Groggy Wakeup - Automated Generation of Power-Efficient Detection Hierarchies for Wearable Sensors

Ari Y. Benbasat[1] and Joseph A. Paradiso[1]

[1] The Media Laboratory, MIT, Cambridge, MA USA

Abstract—**We present a framework for the automated generation of power-efficient state detection in wearable sensor nodes. The core of the framework is a decision tree classifier, which dynamically adjusts the activation and sampling rate of the sensors (termed groggy wakeup), such that only the data necessary to determine the system state is collected at any given time. This classifier can be tuned to trade-off accuracy and power in a structured fashion. Use of a sensor set which measures the phenomena of interest in multiple fashions and with various accuracies further improves the savings by increasing the possible choices for the above decision process.**

An application based on a wearable gait monitor provides quantitative results. Comparing the decision tree classifier to a Support Vector Machine, it is shown that groggy wakeup allows the system to achieve the same detection accuracy for less average power. A simulation of real-time operation demonstrates that our multi-tiered system detects states as accurately as a single-trigger (binary) wakeup system, drawing substantially less power with only a negligible increase in latency.

Keywords— **power-efficient sensing, tiered wakeup, cost-based classification, wearable gait monitoring.**

I. Introduction

Wearable sensor nodes are currently being used in a wide variety of applications. These include, but are certainly not limited to, implantable cardiac defibrillators[1], everyday activity loggers[2] and gait analysis systems[3]. Such systems are part of a new class of sensor-driven applications, leveraging the decrease in both price and size of the components to allow rich, multimodal data streams to be captured by very compact systems. However, wearable sensing applications are constrained to short lifespans by high power usage and limited battery size. For example, long-term medical monitoring is currently hindered by its power consumption. Both fixed environmental sensors – which cannot provide a full picture of an active patient's movements – and body-worn sensors – which require large battery packs and/or frequent replacement – are inadequate.

By concentrating our design efforts on the sensors themselves, rather than on the networking or processing, it is possible to construct sensor systems that achieve their goal(s) while drawing significantly less power. By reducing the power consumption, it is possible to reduce the size of the device and/or increase its lifespan. Both of these parameters directly improve marketability and user acceptance, allowing many more applications to make the transition from laboratory to marketplace and thereby benefit a wider population.

II. Framework overview

The main goal of this work is the reduction of energy usage in wearable sensor systems through the creation and demonstration of new tools and algorithms for the design and construction of power-efficient sensor systems. We start from a fundamental: the *raison d'etre* of these devices is to collect and process data and therefore the design of the sensors should be central. Therefore, we concentrated on reducing the energy usage of the sensors within the nodes. This metric was chosen since both general and tractable, though it is important to note that any power savings in the form of reduced sensing also correspond to further power savings through a reduction in:

- data to process.
- data to transmit or store.
- data to analyze (particularly for a human expert).

Any gains through this work can be considered independently from the large body of work exploring power savings through improvements to *ad-hoc* networking protocols and processor efficiency.

The framework presented is centered on the concept of "groggy", or tiered, wake up. This is in contrast with the more common binary wake up systems, which have only two modes: fully active, collecting all possible data and drawing maximal power, or fully asleep, collecting no data and drawing virtually no power. Instead we envision a system with a number of different levels of activity and associated power usages. Each of these comprises the currently active sensors for state determination and their sampling rate, and algorithms to describe the levels transitions. Our goal is to determine the system state at any given point in time for the smallest outlay of energy. The response thereto is not constrained

The form of the solution is such that the sensor sampling rates, as well as the transitions between them, are generated in a semi-autonomous fashion and can easily be embedded in

hardware. Therefore, the work should be applicable to a wide variety of applications.

Given a desired application, the design process will proceed as follows. Hardware for the individual application will be configured. In the initial deployment for testing and sample data collection, it is assumed that the system will include any sensors that could possibly be of value for state determination. A training data stream is collected and is annotated by the application designer. It is then used to determine the sensor data necessary to differentiate between the states. This information allows the final form of the hardware to be built (possibly with a pared down sensor set) and the state determination algorithm to be implemented on it. Each of these tasks is discussed in the proceeding sections.

III. Related works

There are a number of tiered wakeup systems, mostly built in an *ad-hoc* fashion, which have recently appeared in the literature. A group at UC Berkeley[4] examined the problem of detecting and identifying civilians, soldiers and vehicles traveling through a dense grid of independent sensor nodes. The nodes were awakened using a passive infrared detector, which then activated a microphone and magnetic sensor to identify the source of the trigger. This project did not meet its lifespan goals for a number of reasons. Key among them was the much higher incidence of false alarms than predicted, mostly caused by moving flora. This illustrates the danger of designing systems without the use of real-world sample data streams, instead relying on hand-scripted thresholds[5]. Also, the system draws more power than necessary since it turns on all sensors after a trigger, while the acoustic sensor alone can be used to distinguish between humans and vehicles and draws far less power than the magnetic sensor. A more analytic approach would likely have both identified these flaws sooner and made them easier to correct.

Van Laerhoven[6] designed an activity monitor based on a cluster of tilt switches and a single two-axis accelerometer. The tilt switches are used both for pose detection and to determine when the activity level is high energy to merit turning on the accelerometers. While this design is quite clever, the techniques presented have all been determined and hard-coded for a single application and sensor set. There is no apparent way to extend or generalize these techniques.

Non-state-based (unsupervised) systems have also been reported. Jain and Chang's[7] work on adaptive sampling for sensor networks is one example. They use the innovation of a Kalman filter[8] as a measure of the entropy rate of the data stream and adjust the sampling rate accordingly. Note that this is a purely entropic approach - more data is collected because the phenomena are varying at a faster rate. While this technique generated good results in a sample application, it assumes that data should be collected in all states and cannot differentiate between them. Further, the Kalman filter itself is a fairly structured (and computationally expensive) model that would not be appropriate for all systems.

IV. Hardware

The prototype hardware for this framework is implemented using a modular sensor platform we have designed. This platform is based around a series of circuit boards (or panes), each of which instantiates a specific sensing modality – *e.g.* inertial sensing, tactile sensing or ambient sensing – with the goal of reducing needless reimplementation of common components. Each board encapsulates the best practices in a given field, thereby saving substantial design time. Further, these boards can be arbitrarily combined and recombined, allowing for rapid prototyping and testing of proposed sensor combinations. For a given application, the designer can use this platform to quickly put together a sensor node with which to collect training data. We discuss two important design characteristics below. A more detailed discussion of this platform can be found in [9].

Since much of the power savings of the framework is predicated on power-cycling the various components, reducing the wake up time is key to minimizing the power wasted during that interval. For sensors, this parameter can vary widely both between different sensing mechanisms for a given phenomenon (*e.g.* effectively nil for a phototransistor to 40ms for a IR rangefinder) and individual parts (*e.g.* 8ms for the ADXL202 MEMS accelerometer to 100ms for the pin compatible MXR2312 thermal accelerometer). The wake up time sets the upper limit of how quickly a sensor can be cycled while still offering power savings over continuous activation. It should be noted that the availability of this information is spotty at best – given on some datasheets while completely ignored on others. Further, no information is given about the power draw during wakeup. In many cases it is likely the same as normal, though for some sensors (*e.g.* that need to charge internal capacitor or equilibrate filters) it may well be noticeably more. Further, since most microcontroller-based analog to digital converters can sample far faster than sensors can be activated, the energy used to wake up the sensor can be considered to be the energy used per wakeup cycle.

A second key design technique will be the use of multiple sensors to measure a single parameter of interest. For example, the inertial board uses both passive tilt switches and accelerometers to measure motion. The vast majority of sensor systems limit themselves (usually in the interests of simplicity or compactness) to a single sensor for each modality of interest. No matter how efficient such an implementation is for extracting information, it is guaranteed to be power inefficient

in states were less (or more) data is necessary to determine the transitions. A system which can tailor its sensing in real-time to the current state of the device can draw far less power on average. While it seems counterintuitive that we can make a system more power-efficient by adding complexity (and/or redundancy), the key is that the system has been given a new, lower energy source of information.

V. Pattern recognition

A. Data Collection

We have chosen a supervised training approach based on the assumption of fairly constrained applications and clarity of designer intent. The main benefit of this approach is the ability to ascribe specific meaning to the chosen states (*e.g.* walking), to combine states that might otherwise be separated (*e.g.* fast and slow gaits) and to ignore altogether portions of the data stream that could potentially be considered interesting (*e.g.* skipping).

Training data is collected using the following two-part procedure to acquire the most relevant data. The first stream will be reasonably short and contains the active (high energy/variance) states – both those considered interesting (*i.e.* to be detected) and not – that are known to the designer. The selection of states is left to the discretion of the application designer. The second contains a long-term background recording, to provide a baseline for the uninteresting cases and to catch states that were not considered by the designer above. Both streams are captured at the maximum useful data rate. For wearable/human applications, we use 200Hz.

While a long-term data stream could be used as the sole source of the training data, this tends to be inefficient for a number of reasons. Firstly, the length of the recording necessary to acquire good examples of all complex states can be quite long and their duration may be quite short. Secondly, the vast majority of the uninteresting data collected will be of little to no value in the classifier training.

B. Feature Extraction

A set of simple first order functions is used to calculate the features – the windowed mean, variance, minimum and maximum. These features have been used successfully on time series of human motion, both with inertial[10] and video[11] data. These are good general statistics for two reasons. First, they are algorithmically efficient to implement in a point-wise fashion such that they can be calculated quickly and for little power in an embedded microcontroller. Second, since the data stream will be wide sense stationary over a window size equal to the period of the data (within any given state), the mean and variance will be constant (with the exception of additive noise). This converts a sequence of time varying values to one that varies with state alone.

There are two free parameters in the calculation of these features – window size and sampling frequency. Window size is fixed as the period of the motion of interest, for reasons given above. On the other hand, multiple sampling frequencies are used to allow the classifier to choose the lowest rate (and power usage) that achieves its accuracy goals. However, to avoid having to power cycle a microprocessor on a complex and irregular schedule (which would vary with state), we limit the frequencies to the Nyquist frequency and power of two fractions thereof.

C. Classifier Design

The specific goal of this work is to create a hierarchy of activation states to allow the system to make a state determination using as little information (*i.e.* energy) as possible. Therefore, the classifier used should be able to make decisions in the same fashion - using more or less data as needed. Most classifiers do not have this ability, and instead require that all data be present to make any state determination.

In contrast, decision trees structure classification in the form of a series of successive queries, with each response leading to a following query until a state is determined[12]. In this way, the tree uses different sets of features to classify different states (or subsets thereof). In the case of an unbalanced tree, some classifications are made with fewer decisions (and therefore less energy) than others. Overall, the desire for hierarchical activation requires a hierarchical classifier. Further, the query-based structure allows for very fast implementation in an embedded microcontroller, since only simple comparisons are required to evaluate the classifier (beyond the calculation of the features themselves). Therefore, decision trees with are used in this framework.

In general, our system follows the CART algorithms for constructing decision trees as detailed in [13]. The key difference is that we take the cost of the features – which we define as the energy necessary to collect and calculate them – into account. In the case of sensor data, this almost always reduces to the cost to power up the sensor itself. The hierarchical structure of the decision – where certain sensors may already have been used to answer a query – adds some complexity. Any sensor used at the same (or higher) sampling rate higher in the tree has its cost reduced to zero, since the data has already been collected. If the sensor was used at a lower sampling rate, the cost of use is discount by that already paid, since some of the necessary data is already being collected.

Energy usage is taken into account by reducing the splitting criterion used to determine the query to use at any given node *i* by a factor of $(1+ TC_i)^W$, where TC_i is the test cost for the sensor at that node and W is a parameter used to adjust the trade-off between power and accuracy. This factor does not

distinguish between active sensors and requires unused sensor to achieve a certain quality of split (depending on W) before they will be activated. To avoid non-linearities, test costs should be scaled such that the minimum value is greater than ten. The ratio of the value of the splitting criterion for a perfect split and a meaningless one sets the maximum useful value for W – in the case of the Gini index used in CART, this value is 0.29.

VI. Embedded software

Having trained an appropriate classifier with the desired power and accuracy characteristics, it is fairly straightforward to implement it on a wearable sensor node. The decision tree is encoded in array format, with each entry containing a sensor feature to test on, the threshold for that test, and the index of the next node given a positive or negative result. Entries corresponding to leaf nodes will simply contain the state determination. In either case, a listing of the sensors necessary to get to that point is also given. The code itself keeps track of the currently active sensors as well as an accounting of the recent requests for inactive sensors.

The code execution cycles through a fairly simple loop. The processor awakens at a predetermined time and awakens and collects the data from the active sensors and updates their features. Constant time algorithms for the features are well known and will not be presented here. The classifier itself is then run. If a state is determined, the processor executes the desired response (set by the application designer). If the information necessary to determine a state is not available, the system is considered to be in an indeterminate state. Next, the sensor needs of the tree are examined, with an update of the accounting of both the sensors that are needed but are inactive, and the sensors that are active but are not needed. The system then goes to sleep until the next cycle. Cycle length is determined by the highest update rate among the currently active sensors.

Effectively dealing with sensor activation is key to the efficient operation of this system. Instantly turning on sensors when requested or turning them off when not used to make a decision will leave the system highly susceptible to noise. Waiting too long to activate the sensors will increase latency, while waiting too long to turn off a sensor will waste power. In the current simulations, a sensor must be requested or unused for 5 cycles before its activation state will be altered. This parameter has not yet been optimized. It may be that a higher value is necessary, since whenever a sensor is reactivated there is a delay equal to the window length (or one cycle of the state of interest) before enough data will be available to calculate the features and run the classifier again. Therefore, it may be necessary to sacrifice some power to keep the latency to a reasonable level.

VII. Limitations

We note two important limitations imposed on this framework. First, we have limited the systems examined to the use of passive sensors – those which measure the environment without affecting it. Active sensors (*e.g.* sonar, radar) - those which control the transmission as well as the reception of the measured signal - are excluded both because of their power usage (one to two orders of magnitude greater than passive sensors) and their complex power management (both the output power and the sampling rate can be adjusted individually).

Structurally, any wake-up based system has the potential to miss anomalous events (*i.e.* those with no precursor) - either entirely or during the state determination procedure. While this is a problem in general, it should be minimized given our concentration on wearable sensing. Since most activities in this domain take place on the order of seconds (at minimum) and state determination requires at most a second, the chance of missing an event is minimal.

VIII. Analysis

A. Data Set and Collection

For testing, a data stream containing a wide variety of different ambulatory activities was collected, using a shoe mounted node containing three axes of gyroscopes, three axes of accelerometers and a four-way passive tilt switch. The set of activities chosen was: normal gait, walking uphill and downhill, ascending and descending stairs, and shuffling gait. This data set allows us to create classifiers that attempt to separate a single (complex) ambulatory activity from the rest. Such a system would be valuable for Pakinson's Disease patients, where a doctor would be most interested in collecting information about the frequency and parameters of the patient's shuffling episodes[14]. For patients with total knee replacement, activities such as ascending stairs (where the knee flexion is >90°) are the most important to measure[15]. Such cases allow for far more complex and richer classifiers than those used to simply separate ambulatory from non-ambulatory (roughly: still) states.

Data was collected using the process described above. The active data stream contained two separate segments of approximately 30 seconds in length of each of the non-walking motions. These motions were each bookended by a segment of normal gait. This data was collected in a single session lasting approximately 30 minutes and was recorded with a video camera. Annotations were later added to the data stream based on camera's time stamp.

The long-term data stream was designed to collect data representing the everyday activities of an office-bound worker. Two hours of data were collected with the subject

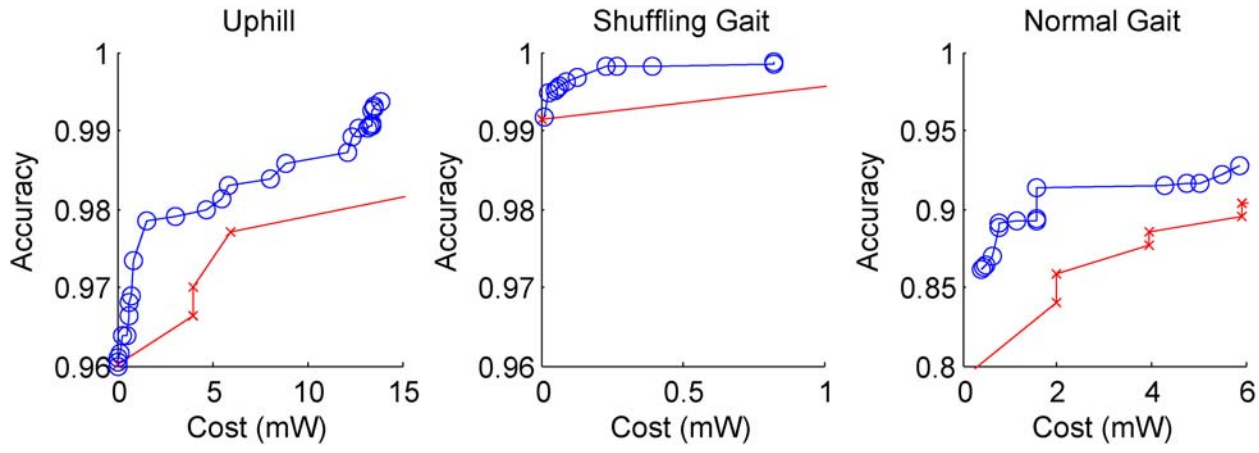

Fig. 1: Power/Accuracy trade-off curves for decision tree and SVM classifiers

sitting at his desk performing a number of basic activities -- typing, reading, searching for papers, *etc.* Non-ambulatory motions such as adjusting the position of the feet, moving from one desk to another and waving the feet under the desk were collected. The subject recorded his activities in a simple diary, with annotations later added to the data stream using the diary as a rough guide and visual inspection of the data stream to more precisely mark the times.

B. Classifier Performance

The above data stream was used to test the ability of our framework to create classifiers that trade off power and accuracy. Classifiers were trained to detect each of the six walking motions amongst the others. Only the active data set was used to avoid placing a large positive skew on the accuracy (since non-walking motions are trivially excluded using the tilt sensors). The sensors and their power usage at the various frequencies used in the classifiers are shown in Table 1. Note that the gyroscopes cannot be power cycled above 33Hz and the accelerometers above 125Hz. It is assumed that ambulatory motion is 80% normal gait and 4% of each of the other gaits. While these are estimates, we note that decision trees are fairly robust to errors in prior probabilities.

Table 1: Power usage of sensors for various frequencies

	25 Hz	50 Hz	100 Hz	200 Hz
Gyroscope	22.5mW	30mW	30mW	30mW
Accelerometer	0.396mW	0.792mW	1.58mW	1.98mW
Tilt Switch	0.41µW	0.83µW	1.6µW	3.3µW

Figure 1 shows the power-accuracy trade-off curves for detecting uphill, shuffling and normal gait (other classifiers omitted due to space constraints). The final point in each graph is from the decision tree grown without power constraint. We note that while the accuracy always increases with power, there is a knee point at which the marginal gain becomes quite low. This data is given in table 2 below.

Table 2: Asymptotic power behavior of trained tree classifiers

	Inflection Point		Maximum	
	Power	Accuracy	Power	Accuracy
Uphill	5.66mW	0.9824	33.82mW	0.9905
Shuffling	125uW	0.9976	1.585mW	0.9985
Normal	5.89mW	0.9272	10.02mW	0.9325

Figure 1 also shows the results of a Gaussian kernel Support Vector Machine (SVM)[16] trained with various subsets of the sensors (always activated). Only the monotonically increasing hull of the points is shown (hence the two point graph for shuffling). While the eventual maximum accuracy (beyond the right-hand edge of the graph) is greater than that of the decision trees, we note that the trees outperform in the low power domain. This confirms the benefits of the tiered structure of the decision tree, even when compared with a (nominally) more powerful technique.

C. Embedded Simulation

Since the tree classifiers were trained without regard to time information, it is important to confirm that they perform well in real-time conditions. The embedded software described above was simulated in MATLAB. Figure 2 shows the results for uphill gait classifier at the inflection point. Both the ground truth (user annotation) and smoothed detection of the classifier (state changes are only acknowledged if their duration is greater than 0.25 seconds) are shown. Table 3 gives the duration of false positives (FP) and false negatives (FN) compared to the true positives (TP) and true negatives (TN) for the tasks above. Total length is 323 seconds. For

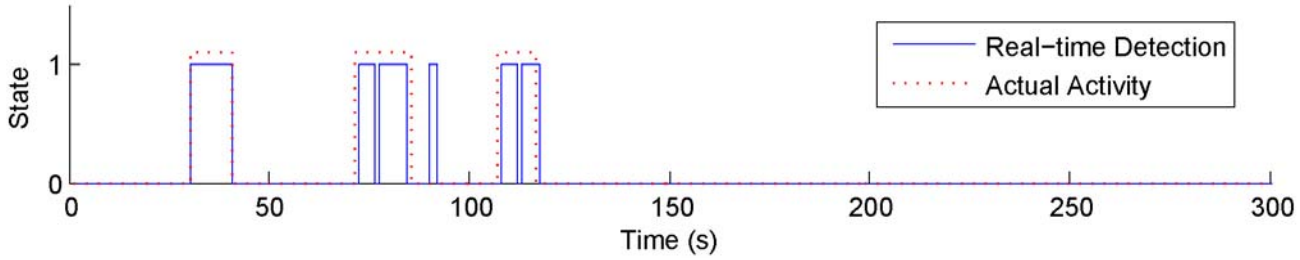

Fig. 2: Model of Real-time Operation for Detection of Uphill Gait

shuffling gait, most of the FPs come for small triggers likely caused by readjusting the feet. For normal gait, most come from overly conservative activity labeling (which is beneficial when training the classifier).

Table 3: Incidence of false positives and negatives on trained trees

	FN	TP	FP	TN
Uphill	5.81s	28.23s	3.11s	285.57s
Shuffling	0.77s	34.93s	15.82s	271.20s
Normal	7.66s	99.62s	50.97s	164.48s

As a final test, we constructed binary classifiers using the root node of the chosen decision trees (*i.e.* if the first query is met, the system is fully activated). This allowed us to estimate the power advantage of a multistage wakeup. For uphill and normal gait, the savings were 56% and 46% (respectively). The best solution for shuffling was found to be a binary classifier.

IX. Conclusions and Future Work

We have described a three-component framework for power-efficient detection in wearable sensors. The first is modular hardware platform for ease of application prototype. The second and key component is a semi-autonomous classifier construction algorithm. Given an annotated data stream, the system uses a version of the CART algorithms, modified to take sensor power into account, to produce a family of decision trees with various power/accuracy parameters. The final component is an embedded implementation of this classifier for use with wearable sensors nodes. Numerical analysis was presented which supports our contention that a tiered wake up approach can reduce power usage with a minimal reduction in accuracy.

Avenues for future work on this framework center around extending from sensor nodes to networks. For the case of a body-worn sensor network, there are two approaches assuming the goal of detecting the current state while minimizing overall power usage. If in any given state one node is the network will be superior to the others, those nodes can share their current state with the other nodes, allowing them to reduce their data collection or turn off entirely. If data from multiple physical locations make some state determinations easier than if done at a single node, the data in question can be shared. In either case, this information can easily be added to the decision tree construction algorithm with a cost based on the energy expended to wirelessly transmit and receive the data.

Acknowledgments

The authors would like to thanks Paolo Bonato for fruitful discussions. The authors also acknowledge the support of the Things that Think Consortium, as well as all the sponsors of the MIT Media Lab.

References

1. J. P. DiMarco. Inplantable cardioverter-defibrillators. New England Journal of Medicine, 349(19): 1836–1847, 2003.
2. Brian Clarkson. Life Patterns: structure from wearable sensors. PhD thesis, MIT, 2002.
3. Ari Y. Benbasat, Stacy J. Morris, and Joseph A. Paradiso. A wireless modular sensor architecture and its application in on-shoe gait analysis. IEEE Sensors 2003.
4. P. Dutta et al. Design of a wireless sensor network platform for detecting rare, random and ephemeral events. IPSN 2005.
5. P. Dutta, University of Calfornia Berkeley. IPSN '05 talk.
6. K. Van Laerhoven, H.W. Gellersen, and Y.G. Malliaris. Long-term activity monitoring with a wearable sensor node. BSN 2006.
7. A. Jain and E. Chang. Adaptive sampling for sensor networks. Data Management for Sensor Networks 2004.
8. A. Gelb, editor. Applied Optimal Estimation. MIT Press, 1974.
9. Ari Y. Benbasat and Joseph A. Paradiso. A compact modular wireless sensor platform. IPSN 2005.
10. Stacy J Morris. A Shoe-Integrated Sensor System for Wireless Gait Analysis and Real-TimeTherapeutic Feedback. ScD thesis, MIT, 2004.
11. Lily Lee. Gait Analysis for Classification. PhD thesis, MIT, 2002.
12. A. Webb. Statistical Pattern Recognition. Wiley, 2002.
13. L. Breiman et al. Classification and Regression Trees. Wadsworth,, 1984.
14. H. Zhu, J. Wertsch, et al. Foot pressure distribution during walking and shuffling. Arch Phys Med Rehabil, 72:390–397, May 1991.
15. D.E. Krebs et al. Biomotion community-wearable human activity monitor: Total knee replacement and healthy control subjects. BSN 2006.
16. R. Duda, P. Hart, and D. Stork. Pattern Classification. Wiley, 2001.