# Spine versus Porcupine: a Study in Distributed Wearable Activity Recognition

Kristof Van Laerhoven and Hans-Werner Gellersen
*Computing Department*
*Lancaster University*
*LA1 4YR Lancaster, United Kingdom*

*{kristof, hwg}@comp.lancs.ac.uk*

## Abstract

*This paper seeks to explore an alternative and more embedded-oriented approach to the recognition of a person's motion and pose, using sensor types that can easily be distributed in clothing. A large proportion of this type of research so far has been carried out with carefully positioned accelerometers, resulting in fairly good recognition rates. An alternative approach targets a more pervasive sensing vision where the clothing is saturated with small, embedded sensors. By increasing the quantity of sensors, while decreasing their individual information quality, a preliminary comparative study between the two approaches looks at the pros, cons, and differences in algorithm requirements.*

## 1. Introduction

The most popular approach for determining a person's posture, motions, and activity is to use external tracking methods that employ cameras, RF beacons, or similar sensors that monitor the body [14] or markers on the body [1]. This method is fairly precise, but also most demanding in terms of setting up the infrastructure, maintaining the hardware and algorithm complexity.

This has especially consequences for the cost and the applicability of such systems: motion capture and tracking platforms are rarely used beyond CAD animation or medical purposes, where precision and absolute position is required. Despite a large interest in tracking people and their activities through distributed sensors, the complexity and reliance on fixed sensors in the environment makes it fairly impractical for daily use.

Nomadic, body worn systems have, in reaction, extensively been researched and argued for in the last decade [7][16]. Having a wearable-only monitoring system that recognizes and records a wearer's daily activities is of substantial value: for instance cross-checking activity patterns versus ECG readings from implanted sensors for wearable monitoring of heart patients [19], automatic creation of diaries listing what the person was doing when and for how long [16], or activity-driven temperature regulation for 'smart' clothing [10]. In general, activity is a valuable component in context aware systems.

From early research prototypes, strapped accelerometers have been used to monitor the motion and orientation of 'points of interest' on the body and to correlate this to activities. Only recently, 'smart textiles'-focused research in wearable computing (e.g., [20]) has offered an alternative vision featuring miniature sensors that are *distributed* and *integrated* in clothing, using weaving structure or the very fibers in the textile as sensors.

This paper argues that embedding and distributing sensors in clothing could be more practical than using strapped-on sensors, and assumes that scores of sensors providing simply binary information could be incorporated and interfaced in clothing. *We aim to investigate how the data from such binary, but highly distributed sensors would perform in an activity recognition scenario.*

## 2. A Comparison in Hardware

The two sensor types that were used in this study are accelerometers (or acceleration sensors) and ball switches (a.k.a. tilt switches). This section introduces both, and compares the requirements and consequences of choosing either sensor as far as the cost in hardware resources is concerned.

### 2.1. The Accelerometer

Aside from automotive applications (e.g., shock and impact detection for airbag deployment), accelerometers can be found in portable and wearable input devices, as well as in a large proportion of wearable sensing research. This sensor can be thought of as a ball that is attached to two springs on opposite sides, and which is placed in a cylinder to limit its movement in two

directions, as depicted in Figure 1a. The output of the accelerometer is in this metaphor the ball's position within the cylinder: shaking the cylinder to the left and right will move the ball's position, but tilting it will do so as well (to a lesser extent). These two effects are referred to as dynamic and static acceleration respectively.

## 2.2. The Ball Switch

The ball switch or tilt switch has historically been popular in pinball and arcade machines as a simple way to prevent players from cheating; the sensor contains a conductive roller ball that closes a switch inside a hollow cylinder when the machine is tilted over a certain threshold (see Figure 1b). The information that this binary sensor provides in terms of orientation and motion is very minimal, but combining several ball switches may boost this sensor's output. Many variations on this type of sensor exist, using gas, mercury or having a slightly different switch mechanism (e.g., a mechanical 'toggle').
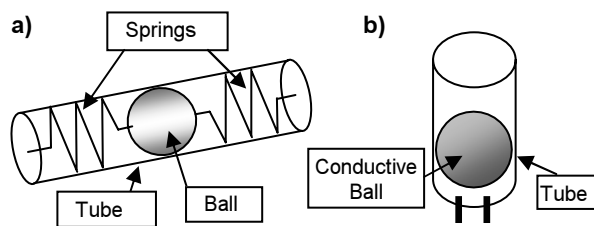


**Figure 1.** The abstract diagrams for the respective models of **a)** an accelerometer and **b)** a ball switch.

## 2.3. Specific Implementations and Comparison

The scope of this paper is not wide enough to perform an absolute assessment into various types of accelerometers and binary tilt sensors. Instead, we offer two prototype sensor-platforms that should be sufficiently characteristic to indicate where the two correspond and differ.

### 2.3.1. Accelerometers: The Spine

The Spine is a progression of a 30-accelerometer outfit constructed earlier [17], and uses the same sensor modules. The new PIC microcontroller (an 18F452 from Microchip) is pin-compatible with the previous 16F877, but operates at twice the clock speed (40Mhz) and has more memory. The number of sensors has been reduced to 20 to allow a faster and more reliable throughput of sensor data. The 2D acceleration sensors (ADXL202JE from Analog Devices) are placed at approximately 10 cm distances from each other, making a total length of the spine around one meter, sufficient to strap to an average arm or leg. The core unit is able to read all accelerometer

values at least 50 times per second via the serial port, which should be more than enough for our purposes; higher speeds can be obtained by changing the serial output modus (i.e., in binary) or the individual accelerometers. The origin of the name should become obvious looking at the two spines in Figure 2 below.
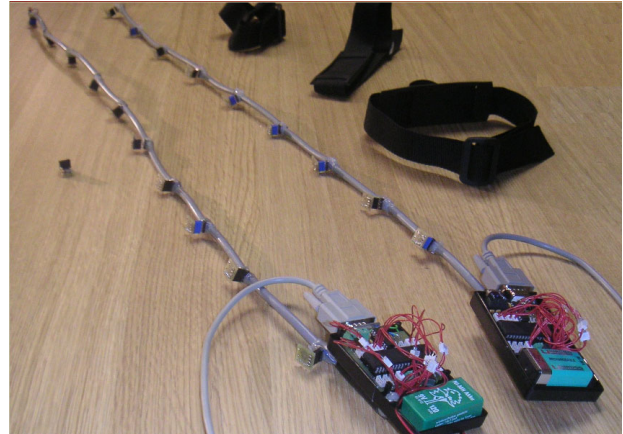


**Figure 2.** Two Spines with their main units open to reveal battery and processing board, straps in the background.

### 2.3.2. Ball Switches: The Porcupine

The ball switch is mainly used independently to reveal whether the object it is attached to is tilted over a certain angle or not. As a switch, it can easily be implemented in circuits to wake up a processor whenever it changes its state, and it needs only a tiny proportion of the accelerometer's power. Size-wise, the classic ball switch is large compared to most accelerometers, but it requires less additional components and could potentially be shrunk to a size below that of the accelerometer.
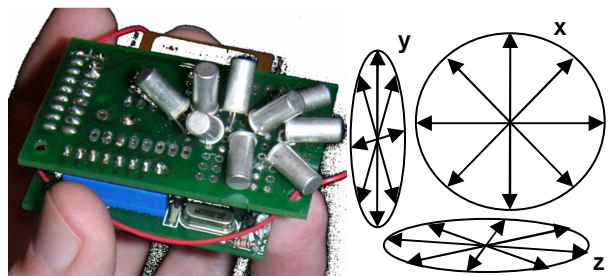


**Figure 3.** One of the first prototypes of the Porcupine, showing how each of the nine ball switches is positioned.

The prototype that was built for this paper's experiment contains nine ball switches that have been placed in 45 degree increments of each other in three perpendicular planes (see Figure 3): in the first plane (X), four ball switches are enough to cover 45 degree increments in all directions (see Figure 4), the second plane (Y) needs only

three additional ones, and the third plane (Z) only two (as the planes overlap). The result is a collection of tilt switches that provide coarse-grained orientation; it is harder to determine whether they are any good at the detection of motion from just the hardware description – this will be part of the investigation in the algorithms section.
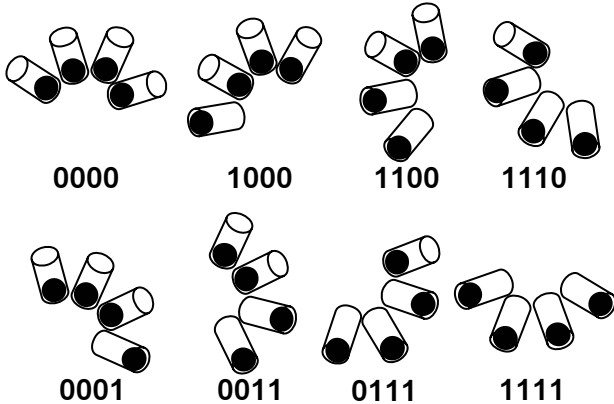


**Figure 4.** How a combination of four tilt switches can give a coarse (45 degree) indication of tilt in one plane; the switch outputs are depicted below the graphs.

The Porcupine (Figure 3, left) is driven by a microcontroller board (running a PIC16F876 from Microchip) with serial and RF communication (BIM2 from Radiometrix) capabilities. In the experiment, the Porcupines were connected to a serial bus running at 115200 baud, through which all sensor data was communicated since mid-air packet collisions limited the wireless output in earlier trials. One acted as master, polling for 3-byte[1] data packets from each of the other nine Porcupines, and forwarding it wirelessly to an RF base station in the immediate area of the experiments.

### 2.3.3. Measurements and Summary

A basic comparison between 1 Spine (20 accelerometers) and 10 Porcupines (90 ball switches) is summarized in Table 1. Advantages of the ball switches lie in their simplicity: they are easy to interface with low-cost microcontrollers and do not need A/D conversions, their power requirements are much lower, and the speeds at which their states can be read on microcontroller level are faster. Moreover, they are often used to 'wake up' microcontrollers in a 'sleeping' mode where it only requires a fraction of its normal power. Accelerometers on the other hand, have a higher resolution for

---

[1] The nine output bits were encoded in twos complement to keep the eventual packet balanced, a necessity in RF communication to guarantee a 'DC free' transmission.

orientation and their signals produce direct motion patterns. Reproducing these numbers can be done with the online building descriptions provided at [15] and [13].

**Table 1.** Basic implementation characteristics and comparison between sensors from Spine and Porcupine.

| Sensing Platform: | Spine 20 accelerometers | 10 Porcupines 90 ball switches |
|---|---|---|
| Maximal current | 40 mA | 22 mA |
| .. for one sensor | 400 uA | 1 uA |
| Battery type | 9V NiMH | 2x 1.5V AA |
| Battery lifetime | 4.3 hours | 2-3 days |
| Time to read all | ~2 ms | ~2 ms |
| Cost (per sensor) | 8 USD | 0.2 USD |
| Size (per sensor) | 50 mm$^3$ | 83 mm$^3$ |
| Extra components | 20 C, 10 R | 90 R |

Both prototype sensing platforms were created specifically for this study, but have since then been used for other studies as well [19]. Although it is not this paper's goal to investigate how both methods perform in terms of implementation, it is interesting to point out that a combination of ball switches might be competitive with accelerometers, especially regarding power consumption.

## 3. A Comparison in Algorithms

In machine learning terms, this paper's main ambition is to balance and analyze the information from a set of scalar sensors and that from a larger set of binary sensors. This will be studied in particular in a scenario of body-centric activity recognition by motion- and orientation sensors.

### 3.1. Experiment Setup

Data from both sensing platforms was logged for a variety of activities. As the experiments are meant to be part of only an indicative study, specifically designed scripts were followed (i.e., no real-life monitoring was done) and only basic activities were considered. Four of the activities were static: Lying, Kneeling, Sitting, Standing, and the other six: Walking, Running, Climbing Stairs, Descending Stairs, Bicycling and Jumping, had typical short patterns of movement.

These activities were chosen because of (1) their presence in related work (e.g., [4][7][9][16]) and (2) their repetitive nature. The former motivation hints at the proposed scenario where these types of activities might be valuable, whereas the latter aims towards a particular design of algorithms, where prototypical data to be

classified is expected to be temporally simple and repetitive, but complex in dimension (multiple sensors contributing to the classification). All sensors were strapped to the legs of the test subjects, primarily because the size of the utilised prototypes did not allow any embedding into clothes; we argue that this has little effect for these early studies.



**Figure 5.** Test subjects were asked to perform certain basic activities from a set script, such as climbing stairs or bicycling - scenes from the Spines' data logging.

To create a generic dataset (i.e., one that can also be used for future experiments), test subjects were asked to wear the two platforms, and follow the experiment's script several times per platform in order to have sufficient data (especially for the Porcupines' switches). Figure 5 illustrates how the data was captured, using two Spines in a strapped-on setup similar to previous studies. Figure 6 shows similar scenes using Porcupines.



**Figure 6.** The activities were chosen so that their data was either practically static (e.g., standing), or contained repetitive patterns (e.g., walking) - scenes from the Porcupines' data logging.

The next sections will explore whether the logged sensor data contains enough information to distinguish these activities and, if so, how easy it is to extract them using algorithms.

## 3.2. Algorithm Overview

The algorithms in this section will be grouped per type, and more attention will be given to algorithms that might possibly perform better for the high-dimensional binary data that the Porcupines' ball switches produce.

Before going into specific algorithm descriptions, though, it is already possible to describe some issues and characteristics we can expect regarding the ball switches' data. Ball switches first of all tend to show 'bouncing' behaviour when they are about to tilt: although a Porcupine may be perfectly still, some of the switches could still alternate between the zero- and one states, providing possibly serious noise. This problem extends to any binary sensor: noise has a more damaging effect on a per sensor basis. Figure 7 shows example data from one Porcupine to illustrate this.

A second issue is the lack of characteristic peaks or other features that can easily be extracted from the Porcupine data. Figure 8 shows a typical set of acceleration time series from 'climbing stairs', which is almost trivial to classify when certain characteristic peaks are extracted for this activity. This type of pre-processing is less obvious with binary data. Take the clustering of sensor data for instance: calculating an average position between binary objects is rarely done. Rather than calculating centroids, a bit string that minimizes the sum of the distance to all objects, called medoid, is used.
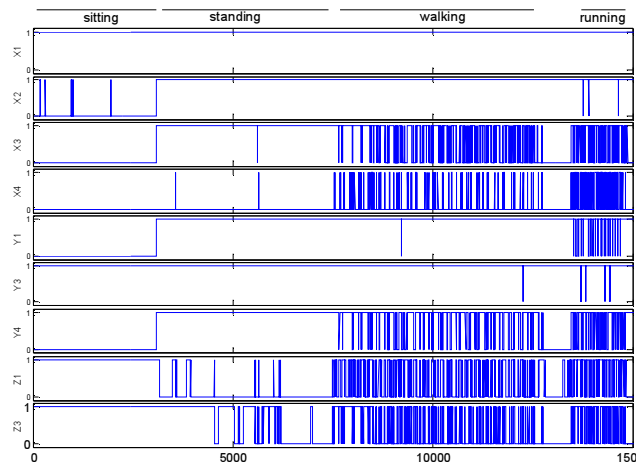


**Figure 7.** Example data from a knee-worn Porcupine: output values (Y axes) versus samples (X axis, ~50 Hz). Notice the noise during the sitting and standing contexts.
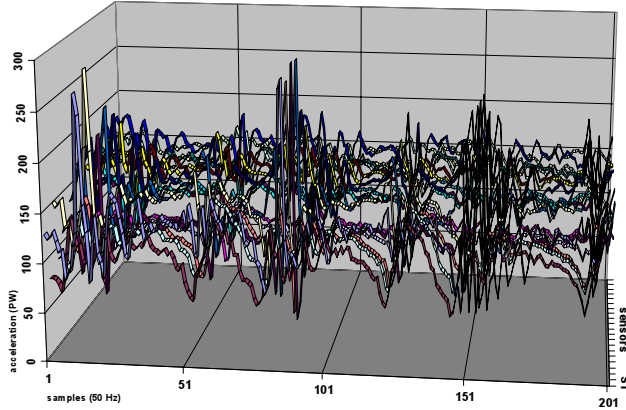
**Figure 8.** Example data from the right-leg Spine for climbing stairs: output values (Y axes) versus samples (X axis,~50 Hz) for all sensors (Z axis). Note the characteristic 'step-up' peaks for the first sensors (S1).

### 3.2.1. Topographic Mapping-based classification

For accelerometer-based platforms such as the Spine discussed in this paper, previous research has been focused on testing machine learning strategies on their data. The proposed algorithms include Gaussian modelling [7], Support Vector Machines [11], Bayesian Classifiers [8], and Kohonen Self-Organising Maps [16], to name but a few. The preparation and choice of features is often a significant part of these studies.

We will confine the accelerometer side of the comparison to an algorithm based on that discussed in [16], using a hierarchical topographic mapping approach (the Kohonen Self-Organising Map) with basic statistics and peak set descriptors as features. This method has shown its merit before, and will serve as a typical algorithm developed for a potentially large amount of accelerometer data.

### 3.2.2. Spiking Neural Nets, Pulsed Neural Nets

Spiking [6] or pulsed [12] neural networks are using one of the most recent, and more biologically plausible, models of artificial neurons, working with action potentials, or spikes, and their *timings*, rather than scalar values. Whereas earlier artificial neurons (often based on the traditional McCulloch-Pitts model) combine continuous inputs in a weighted sum and activation function, spiking neurons use a so-called leaky integrator that decreases with time, and increases when it receives a pulse (see Figure 9 for an illustration).

It is our hypothesis that for the last six dynamic activities (those with recurring motion patterns) the temporal patterns of the switching itself and the knowledge of

which ball switch was triggered are enough to characterise the activities – if the distribution and data acquisition of the ball switches are sufficient. This scheme, where a spike is created each time a binary sensor's state switches, connects seamlessly to a spiking neural network's architecture.
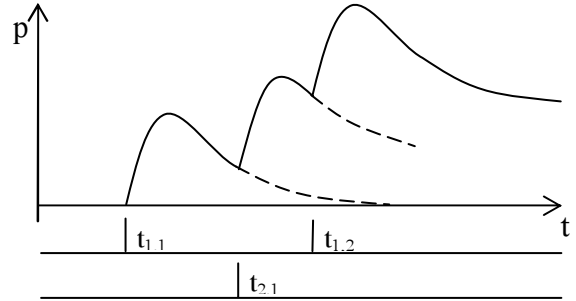


**Figure 9.** Typical leaky-integrate-and-fire behaviour of a spiking neuron: the action potential increases each time a spike arrives, and 'fires' when a certain threshold is met. The time series plot the growing potential p over time for the two inputs below the graph (spiking 3 times in total).

In particular, we have used an unsupervised learning rule for these neurons that modifies the weights that connect the inputs to neurons in a single output layer, slightly similar to the Topographic Self-Organising Map implementation in 2.3.1.

### 3.2.3. Boosting, AdaBoost

Boosting is a type of learning strategy that fits very well with data coming from a distributed sensing system: It concentrates on *combining weak classifiers* to come to a classification scheme that usually has a better performance than each of the classifiers individually. Instead of trying to design a learning algorithm that is accurate over the entire classification space, the focus is more on finding weak 'rule of thumb' type of algorithms that need only to be better than random.

We applied a multiclass generalisation of AdaBoost, one of the most popular implementations for boosting [5] that is traditionally restricted to binary classification, by training for each activity separately. The boosting approach is appealing in our case since we can distribute the weak classifiers locally (i.e., on the Porcupines), and have them send their classification hypotheses, rather than the sensor data, to a central post. We therefore chose 10 classifiers that estimated the overall activity from the 9 local ball switches only. In brief, our AdaBoost implementation assigns and updates weights to each classifier per activity, and the final verdict is gotten through a vote amongst the weighted results.

## 3.3. Results

The logged data was manually annotated in two passes: one for marking the data that would be applicable for providing the ground truth, and a second time for extracting ground truth data that would be used for training the classifiers. The latter set was about 25 percent of the initial ground truth set.

The first results are from accelerometer data analysed by the method described in 2.3.1. The results are plotted in the first confusion matrix in Figure 10. It is not surprising to see that the classifier performs well on all activities, given the amount of pre-processing (especially the peak extraction) that evolved from previous research. The merit of these features becomes clearer when characteristic peak sequences for particular sensors are plotted against each other for the more complicated activities (see Figure 11): though close, they are distinctive enough to end up in different clusters, and thus in different classes.
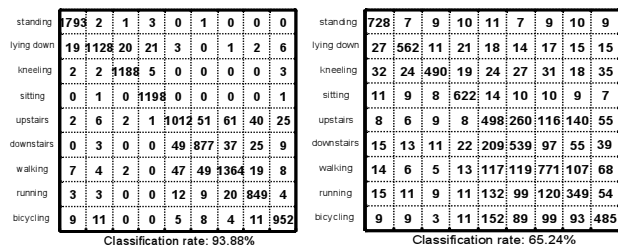
| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| standing | 1793 | 2 | 1 | 3 | 0 | 1 | 0 | 0 | 0 | standing | 728 | 7 | 9 | 10 | 11 | 7 | 9 | 10 | 9 |
| lying down | 19 | 1128 | 20 | 21 | 3 | 0 | 1 | 2 | 6 | lying down | 27 | 562 | 11 | 21 | 18 | 14 | 17 | 15 | 15 |
| kneeling | 2 | 2 | 1188 | 5 | 0 | 0 | 0 | 0 | 3 | kneeling | 32 | 24 | 490 | 19 | 24 | 27 | 31 | 18 | 35 |
| sitting | 0 | 1 | 0 | 1198 | 0 | 0 | 0 | 0 | 1 | sitting | 11 | 9 | 8 | 622 | 14 | 10 | 10 | 9 | 7 |
| upstairs | 2 | 6 | 2 | 1 | 1012 | 51 | 61 | 40 | 25 | upstairs | 8 | 6 | 9 | 8 | 498 | 260 | 116 | 140 | 55 |
| downstairs | 0 | 3 | 0 | 0 | 49 | 877 | 37 | 25 | 9 | downstairs | 15 | 13 | 11 | 22 | 209 | 539 | 97 | 55 | 39 |
| walking | 7 | 4 | 2 | 0 | 47 | 49 | 1364 | 19 | 8 | walking | 14 | 6 | 5 | 13 | 117 | 119 | 771 | 107 | 68 |
| running | 3 | 3 | 0 | 0 | 12 | 9 | 20 | 849 | 4 | running | 15 | 11 | 9 | 11 | 132 | 99 | 120 | 349 | 54 |
| bicycling | 9 | 11 | 0 | 0 | 5 | 8 | 4 | 11 | 952 | bicycling | 9 | 9 | 3 | 11 | 152 | 89 | 99 | 93 | 485 |
| | Classification rate: 93.88% | | | | | | | | | | Classification rate: 65.24% | | | | | | | | |

**Figure 10.** Confusion matrices for the Spine (left) and Porcupine (right) datasets. The cells show the amount of positives per activity (with the true positives diagonally), the classification rates show the rate of true positives over all activities.
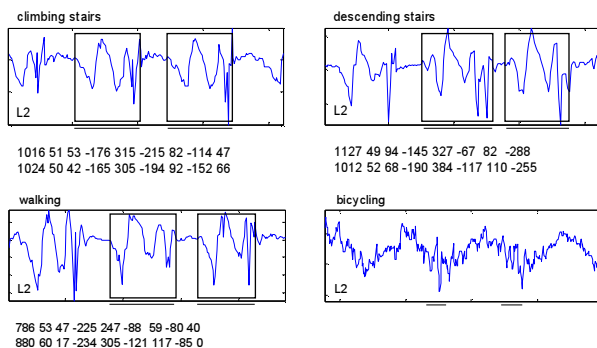


**climbing stairs**

L2

1016 51 53 -176 315 -215 82 -114 47
1024 50 42 -165 305 -194 92 -152 66

**descending stairs**

L2

1127 49 94 -145 327 -67 82 -288
1012 52 68 -190 384 -117 110 -255

**walking**

L2

786 53 47 -225 247 -88 59 -80 40
880 60 17 -234 305 -121 117 -85 0

**bicycling**

L2

**Figure 11.** Peak characterisation for one accelerometer (L2): the numbers underneath the graphs represent the total area, the length, and the areas of the individual peaks of the last two detected peak sets. None were detected for bicycling in this case. The underlined areas mark the last two candidate peak sets.

The results from the proposed spiking neural network (see Figure 10, second confusion matrix) show potential for improvement, but they are reminiscent of early work on accelerometer-based platforms. Keeping in mind that no special features were calculated, and that the raw switches were fused straight into the algorithm, we believe that this approach could have benefits in on-line and embedded algorithms. Especially for the static poses, it performs adequately, given the amount of noise present for individual ball switches (e.g., as in Figures 7 and 12).

After a more thorough inspection of the data, however, it was noticed that no particular characteristic spike trains were generated: deciding factors seemed rather to be down to simple variance in the on-off switching. This indicates that even between our dataset's more complex activities, such as climbing or descending the stairs, different ball switches were triggered, or they were triggered at distinct enough speeds for most of the time.

To make this more apparent, a second test with the same topographic map algorithm as for the accelerometers was performed, but with *just the average and variance* over the last 30 samples as features. Remarkably, the results were comparable to those from the spiking neural net. The results from the AdaBoost approach were also in line with these observations, with an overall classification rate of 62%, using the binary variant of distance-weighted k nearest neighbours as classifiers. The features used were as close as possible to the average (the state most frequently occurring, 0 or 1, during the past 100 values) and the variance (a counter of all the 1s of the past 100 values) per sensor.

The time complexities and algorithm requirements for both approaches were not really investigated here. It is obvious that the performance of the algorithms is sufficiently affordable for the high-dimensional binary data in terms of time-complexity and memory requirements: one sample in both cases takes 90 bits for the ball switches, while it requires at least 400 bits for the accelerometers. Since the individual channels that need to be fused contain low-resolution data, algorithms need less memory and time: the algorithms that worked on the accelerometer datasets required significantly more time, mainly due to the pre-processing.

One key disadvantage for the balls switches' approach that became obvious during the analysis of the experimental data was the lack of visualisation. Just having binary states makes inspection and manual post-annotation of datasets a challenging task (as for example in Figure 12).
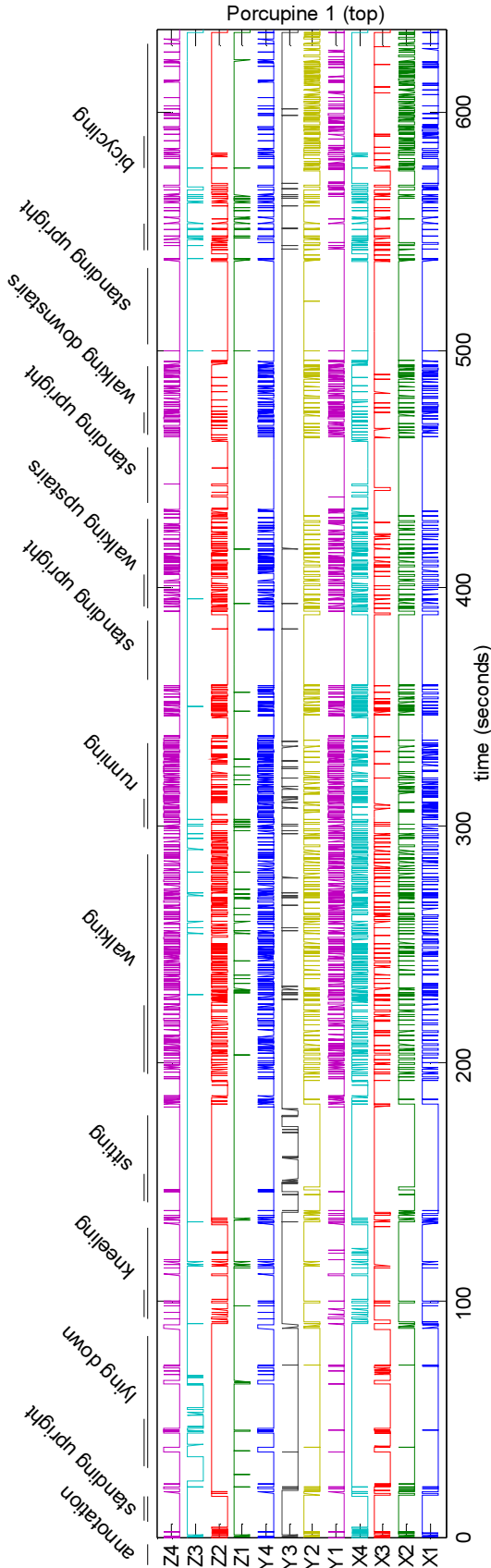
**Figure 12.** Visualisation for annotated data from the top (hip-worn) Porcupine, plotted as switch state (on/off) versus time. All four switches per plane are included to facilitate backtracking of sensors from signals. (Note that this leads to redundancy for X2 and Y2, X4 and Z2, and Y4 and Z4).

## 4. Applications and Outlook

The main objective of this work was to investigate the consequences of taking distribution of 'dumb' sensors a step further. It is in that sense important to stress that one should not over-interpret this study as an advocacy for the use of ball switches in activity recognition. Likewise, binary sensors may indeed have practical applications in integrated fabric sensing, but our core motivation for choosing binary sensors was the consequences their data have for sensor fusion algorithms.

Future plans do include extended hardware prototypes that combine low-power microcontrollers with miniature ball switches, reducing the overall size of the Porcupines, and enabling embedding in clothing and scaled-up distributed networks; a next version is set to be almost a tenth smaller than the current prototype. Simultaneously, microcontroller implementations for the discussed algorithms are underway as well.

Given that both types of sensors are similar in concept, it could be argued that the ball switches could have been simulated from accelerometer data. This is not entirely certain, however, and simulation of the ball switches would have weakened effectively any data comparison.

The most interesting outcome of the experiments was the apparent lack of need for complex pre-processing or algorithms for the Porcupines' dataset. Although it was hypothesised that feature information could be hidden in patterns of the switch signals, no evidence for this was found in our dataset; a higher sampling rate or different types of ball switches might be required to confirm this.

It also has to be stressed that this first dataset is too small to be dependable. In order to generalise from this study, more and longer-term datasets need to be recorded. It is in this regard also interesting to speculate how well algorithms and such a large distribution of binary sensors would cope with problems such as concept drift, where the activities would slowly change with relation to the sensors (due to tiredness, shifting clothes, etc.): a wide distribution might be an advantage in this case.

There are a number of other ways to advance this study. We are exploring scalability issues for classification algorithms and cross-usage of training data over multiple people (i.e., generalisation concerns). Widening the study to other classification problems, such as explicit gesture recognition or activity prediction, would be a further area of future work. The datasets that this research has been (and will be) generating are available for download via the CommonSense website [3].

## 5. Conclusions

Our aim was to study an approach towards activity recognition inspired by the 'smart clothing' paradigm, i.e., under the assumption that the sensors are high in number and heavily distributed throughout the fabric, yet individually providing only a tiny chunk of information. Compared to the more classic methods of placing sensors where they are absolutely needed, we argue that this approach would be more suitable for scenarios where strapped sensors might be obtrusive, or where optimal sensor location is hard to discover beforehand (e.g., in skirts or dresses).

Two experimental prototypes were introduced to demonstrate and compare the traditional and proposed approaches. Although the experiments do not constitute any actual proof of theory, they do indicate that distributing sensors - even while heavily neglecting their accuracy - could be valuable and that more, distinctive, research is needed in algorithms for simple sensor fusion.

## Acknowledgements

## References

[1] 3DEqualizer: http://www.3dequalizer.com/

[2] O. Cakmakci, M. Koyuncu and M. Eber-Koyuncu. "Fiber Computing". In *Proc. of the Workshop on Distributed and Disappearing User Interfaces in Ubiquitous Computing*, CHI 2001, Seattle, WA, USA.

[3] CommonSense: http://ubicomp.lancs.ac.uk/commonsense

[4] J. Farringdon, A. Moore, N. Tilbury, J. Church and P. Biemond. "Wearable Sensor Badge & Sensor Jacket for Context Awareness". In *Proceedings of the Third International Symposium on Wearable Computers* (ISWC'99), San Francisco, pp.107-113.

[5] Y. Freund and R. E. Schapire. "A Decision-Theoretic Generalisation of On-line Learning and an Application to Boosting". In *Journal of Computer and System Sciences*, 55(1), pp.119-139, August 1997.

[6] W. Gerstner, and W. Kistler. "Spiking Neuron Models". Cambridge University Press. 2002.

[7] A. R. Golding and N. Lesh, "Indoor navigation using a diverse set of cheap, wearable sensors". In *Proceedings of the third International Symposium on Wearable Computers*, pp. 29-36. 1999.

[8] N. Kern and B. Schiele. "Multi-Sensor Activity Context Detection for Wearable Computing". In *Proc. Of European Symposium on Ambient Intelligence*, 2003, Eindhoven, The Netherlands.

[9] N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Tröster. "Wearable Sensing to Annotate Meeting Recordings". In *Proceedings of the Sixth International Symposium on Wearable Computers* (ISWC'02), pp. 186-196, Seattle, 2002.

[10] K. Kukkonen, T. Vuorela, J. Rantanen, O. Ryynänen, A. Siili, and J. Vanhala. "The Design and Implementation of Electrically Heated Clothing." In *Proceedings of the Fifth International Symposium on Wearable Computers* (ISWC'01), Zurich, 2001.

[11] G. Loosli, Canu, S. and Rakotomamonjy, A. 2003. Détection des activités quotidiennes à l'aide des séparateurs à Vaste Marge. RJCIA, France, pp. 139-152.

[12] W. Maass and C. M. Bishop. "Pulsed Neural Networks". The MIT Press. 2001.

[13] The 'Porcupine' prototype building instructions: http://ubicomp.lancs.ac.uk/~kristof/research/notes/porcup

[14] Y. Song, X. Feng and P. Perona. "Towards Detection of Human Motion". In *IEEE Proc. of Conf. Computer Vision and Pattern Recognition*, vol I, pp. 810-817, Hilton Head Island, South Carolina, June, 2000.

[15] The 'Spine' prototype building instructions: http://ubicomp.lancs.ac.uk/~kristof/research/notes/spine

[16] K. Van Laerhoven and O. Cakmakci, "What Shall We Teach Our Pants?". In *Proceedings of the Fourth International Symposium on Wearable Computers* (ISWC 2000), Atlanta, 2000.

[17] K. Van Laerhoven, A. Schmidt and H.-W. Gellersen "Multi-Sensor Context Aware Clothing". In *Proceedings of the Sixth International Symposium on Wearable Computers* (ISWC'02), Seattle, 2002.

[18] T. von Büren, P. Lukowicz, and G. Tröster. "Kinetic Energy Powered Computing – an Experimental Feasibility Study". In *Proceedings of the Seventh International Symposium on Wearable Computers* (ISWC'03), pp. 22-24. New York, 2003.

[19] UbiMon: http://www.ubicare.org/projects-ubimon.shtml

[20] R. Wijesiriwardana, T. Dias and S. Mukhopadhyay. "Resistive Fibre-Meshed Transducers". In *Proceedings of the Seventh International Symposium on Wearable Computers* (ISWC'03), pp.200-209. New York, 2003.