Continuous activity recognition in the kitchen using miniaturised sensor button

Jamie A Ward[1], Nagendra B Bharatula[1], Gerhard Tröster[1], Paul Lukowicz[2]

(1) Wearable Computing Laboratory, ETH Zürich
8092 Zürich, Switzerland, www.wearable.ethz.ch
{ward,bhargava}@ife.ee.ethz.ch

(2) University of Passau
Innstrae 43, D-94032 Passau, Germany
paul.lukowicz@uni-passau.de

***Abstract*— In this paper we present and evaluate a miniature, low power wearable platform for continuous activity recognition. By continuous recognition we mean that we can spot relevant activities in an unsegmented stream of sensor data containing a considerable amount of '*NULL* class' (arbitrary other activities performed by the user). The main contribution of this work is to show that such continuous recognition in a complex environment is possible using an unobtrusive device that, including battery and housing is the size of a standard watch and can operate for around 25 hours without re- charging. This is a perquisite for a practical implementation of a wide range of applications such as assisted living, assembly support and adaptive interfaces.**

The device, having a thickness of 11 mm and a radius of 15.5 mm (plus 3 X 20 x 20 mm for the battery), makes use of sound and acceleration information, recorded from the user's wrist, to recognise a selection of activities involving the hand. Building on previous work by our group we fuse the output of two separate classifiers: one using sound and one using acceleration information to distinguish relevant events from the *NULL* class.

To evaluate the system eight subjects were asked to perform five repetitions of a predefined sequence of activies (or 'recipe'), each lasting between three and five minutes (a total of 138min of continuous data). Disregarding timing issues our system can achieve a frame by frame error rate as low as 14%.

I. Introduction

The ability of a device to monitor and model the user's action and the situation in the surrounding is considered to be one of the key features of future generation smart appliances and mobile devices. Referred to as context awareness it enables systems to automatically adjust the functionality and configuration to the specific needs of the user at a given moment (Abowd *et al.* [1]). The three main approaches to achieving context awareness are: (1) video/image analysis (e.g. Starner and Schiele [16]), (2) the use of sensors integrated in the environment and (3) the use of wearable sensors mounted on the user. This paper deals with context recognition using wearable sensors.

A major thrust in wearable context recognition research is the use of sensors mounted on the user's hand and arm. In extensive previous work by the authors of this paper, it was shown that it is possible to recognise a subset of activities in a carpentry scenario using body mounted accelerometers and microphones [19], [13]. These methods were also shown to work using only wrist-worn sensors [17].

In terms of practical usability of a context recognition system the wrist location is particularity attractive since people are used to wearing watches. However such systems are only viable if:

1) The inclusion of context recognition does not increase the size and weight of the device beyond what is acceptable for a standard watch
2) Even operating in a continuous mode the device must be able to run for days without the need for re-charging.

Initial work targetting the above considerations has been performed by Krause *et al.* [2]. They have optimised the recognition of simple activities such as walking sitting, standing from an accelerometer signal to run for several hours on a wristwatch computer platform.

The work presented in this paper goes beyond both our previous work and work by other groups in the following way:

1) We address the spotting of complex activities such as the individual steps of the cooking process.
2) We use not just an accelerometer but also a microphone.

3) We present the implementation of a complete button size, ultra low power platform specifically tailored for low power context recognition using our methods.
4) We have implemented all our algorithms to perform on-line real time recognition on this platform. While previous work by our group has used accelerometers and microphones for low power recognition, such on-line, on-device implementation has not been previously achieved.

As a validation of our system an eight-subject dataset was collected using the wrist-worn device and used to evaluate the recognition algorithms in two steps: first we performed a 'best-case' evaluation using Matlab. Disregarding timing errors we arrive at a frame by frame error rate of 14%. We then did a full evaluation using the IAR-workbench MSP simulation using the exact code that runs on the wrist-worn device. This returned a frame by frame error rate of 21%.

While a large error by some standards (e.g. compared to speech and gesture recognition systems), considering the complexity of the task and the extremely low resource consumption this is a reasonably satisfactory results. We also believe that it is sufficient for many (certainly not all) applications. Thus for example in many nursing and assisted living scenarios dealing with cognitive disorders the interesting parameter is the long term trend of the daily routine of a person (erratic deviations are a often a sign of emerging problems). In such cases a larger frame by frame error rate should not be a major problem in spotting such trends.

A. Related work

Westeyn *et al.* [20] used wearable accelerometers to spot certain behavioural activities in autistic children, and Bao and Intille [4] to recognise multiple full-body activities. Chambers *et al.* [9] investigated the recognition of certain Kung Fu moves by augmenting visual recordings with wrist-worn accelerometer data. Of more intricate hand activities, such as interaction with objects, or gesticulation, there have been several works using accelerometers - generally involving sensors either on the objects being manipulated, as presented by Antifakos *et al.* [3], or embedded in special gloves, as shown by Fang *et al.* [11].

Staeger *et al.* [15] introduced a method of recognising daily activities using sound. Pelton *et al.* [12] investigated the use of sound for analysing situations, such as detecting which location the user is in - bedroom, street, church, etc. Büchler [8] presented a method of using sound analysis to improve the performance of hearing aids. More recently, Scott *et al.* [14] used sound for fine grained location detection within a building. And in [10], Chen *et al.* used sound - rather distastefully for some - to detect activities in a bathroom.

Towards the vision of a wearable sensor node that is small enough to be integrated into clothing, yet powerful enough to be useful for activity and context recognition, previous work at ETH has dealt with a number of hardware issues such as suitable electronic packaging [5], power and size optimisation for multi-sensor context recognition [6] and development of hybrid micro power supplies [7].

II. THE SENSORBUTTON HARDWARE

The architecture of the *SensorButton* is illustrated in Figure 1 and its features are summarised in Table I. It is composed of 4 main parts: digital, analog, RF and power.

Digital: the *SensorButton* is centred around a 4 MHz 16-bit MSP430F1611 micro-controller (Texas Instrument). This micro-controller has 48KB of program Flash memory, 256 bytes of data flash memory and 10KB of data RAM. Though it requires a relatively small amount of power (2.4mA at 4MHz and 3V), the MSP is a full 16-bit micro-controller and includes a hardware multiplication unit (e.g. for efficient multiply-accumulate operations typical of digital signal processing applications).

The micro-controller is in charge of collecting data and processing them (e.g. doing data classification or compression). It has a 12-bit ADC converter with 8-input channels that is used to sample analog sensor inputs.

The micro-controller also communicates with other *SensorButtons* or with a base station (e.g. a desktop computer), either with the wireless link or over an RS-232 serial line. An external extension board acts as an RS-232 to USB interface so that a computer with an USB port can communicate with the micro-controller over a virtual COM port. This extension board may also be used power the *SensorButton* with the power line derived from the USB port. A JTAG interface allows to program the micro-controller.

Analog: the *SensorButton* contains three sensors useful for many wearable computing scenarios.

A 3-axis accelerometer provides acceleration information that can be used in wearable systems to detect and classify user motion (e.g. in this work the detection of arm movement).

A MEMS microphone is also built in. This allows the *SensorButton* to sample and process audio data.

The *SensorButton* has a light sensor that allows environmental lighting to be used as an additional source

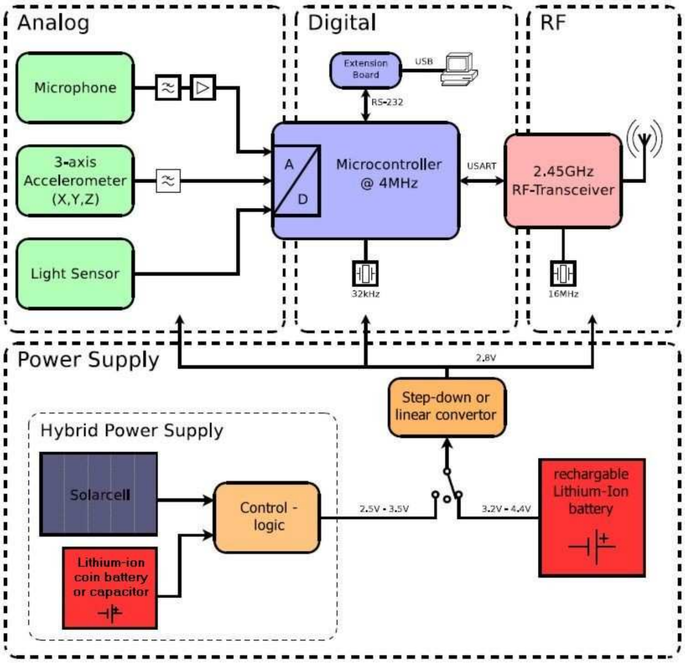

Fig. 1. Architecture of the SensorButton

of information for wearable computing scenarios, e.g. to detect day from night, or detect the hand going in a pocket (light occlusion). In this work however, the light sensor is not used.

All the sensors are filtered using a second order Butterworth filter and are connected to one of the analog inputs of the ADC converter of the micro-controller.

RF: the *SensorButton* is fitted with a 2.45GHz shock burst transceiver (nRF2401E, Nordic Semiconductors) for wireless data transmission. This has a low power consumption of 26 nJ/bit. The 2.4GHz frequency band allows the use of a compact patch antenna (12.5mm x 4mm).

Power: the *SensorButton* is powered by a lithium-ion battery (130mAh, 3.7V) with a step-down converter (TPS62220, Texas Instruments). An external power-supply can be selected by toggling a miniature switch (ESE157, Panasonic).

To reduce power consumption, the power to the three sensors can be individually toggled by the micro-controller via a CMOS analog switch (MAX4783, Maxim).

We have also investigated a hybrid power supply composed of a solar cell and a miniature battery for very low power applications up to 1 mW. (This is not used in the current work however).

- 4 MHz 16-bit MSP430F1611 micro-controller (Texas Instruments)
- 48 KB Flash program memory
- 256 bytes Flash data memory
- 10 KB data RAM
- Battery: lithium-ion, 130mAh, 3.7V (LPP402025, Varta)
- 32KHz external clock feeding a digitally controlled oscillator at 4MHz
- Wireless 1Mbps 2.45GHz transceiver (nRF2401E, Nordic Semiconductors)
- 3 axis accelerometer (LIS3L03AQ, ST Microelectronics)
- Light sensor (SFH3410, OSRAM)
- Microphone (SPO1013, Knowles Acoustics)
- JTAG programming interface
- Serial line for data communication
- Physical characteristics: 31mm diameter, 11mm thickness, 12g weight

TABLE I

CHARACTERISTICS AND FEATURES OF THE *SensorButton*.

A. *Electrical and mechanical characteristics*

Implementation-wise the *SensorButton* is composed of two stacked PCBs. The overall system size is 31mm diameter and 11mm thick (7.2mm when PCBs are glued together instead of stacked with connectors). The total weight is 12g (including battery, excluding plastic casing). Figure 3 shows the assembled *SensorButton*. Figure 2 shows the two PCBs making up the device.

The micro-controller, the battery and serial I/O line are located on the bottom PCB. The sensors (accelerometers, light sensor and microphone) and RF link are located

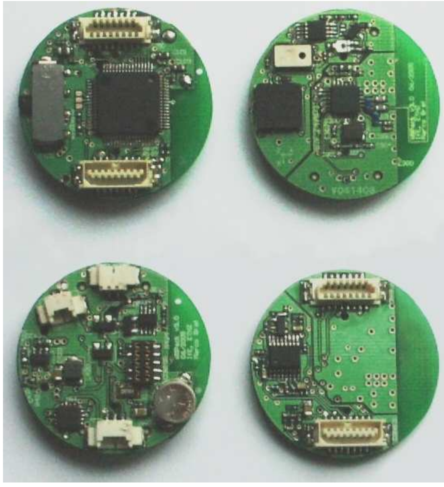

Fig. 2. *SensorButton* PCBs. The bottom PCB (left column) contains the micro-controller and power-supply. The top PCB (right column) contains the sensors and the RF transceiver.

Fig. 3. The complete SensorButton and power supply, mounted as a wrist-worn unit

on the top PCB. Both PCBs have 4 layers: the top and bottom layers with components, an internal ground layer and an internal signal layer.

In order to reduce EMC (electromagnetic compatibility) problems the following steps were followed:

- The system is partitioned in digital, analog and RF parts.
- The length of tracks carrying high currents are kept short.
- Tracks that could influence each other are separated.
- Decoupling capacitors are used whenever necessary.
- Analog power is separated from the digital power by a CLC filter.

Two connectors are used to stack the boards and provide a stable mechanical connection between the two boards besides the electrical connection. Analog signals and power run on one connector, while the digital signals for the RF-transceiver run on the other. The three power lines that can be toggled by the micro-controller are located on the analog connector, as well as the analog signals. Analog signals include the sensors (accelerometers, microphone and light sensor) and battery voltage and optional solar cell voltage for monitoring purposes.

The stacked approach offers a lot of flexibility: the sensors or the wireless communication device may be changed by simply redesigning the top PCB while the bottom PCB is left unchanged. Spare I/O lines may also be used to seamlessly add new sensors to the *SensorButton*.

III. Experiment

The kitchen scenario introduced in this paper uses 8 subjects (3 female and 5 male). All subjects were right handed and the same tools and workplace were used throughout. The subjects were asked to follow a recipe involving 12 different kitchen activities: scrubbing vegetables under a running fawcet, peeling an apple, taking objects from a drawer, grating a carrot, slicing an apple, using an electric blender, pouring from a jug, using an electric hand mixer, squirting lemon from a dispenser, stirring with a fork, and cutting bread. Each of these activities were separated by periods of *NULL* - moving around, picking up items, standing idle, etc. The procedure, shown in Table II was repeated 5 times for all but one of the subjects (due to various time and technical issues that subject, a male, only managed 3 repetitions).

The entire dataset was collected using a single wrist-worn SensorButton. A device was sewn to a fabric sleeve and worn on the right wrist of each subject (the plastic housing, shown in Figure 3, was a later addition). Raw data from the 3-axis accelerometer, sampled at 109.9 Hz, and the microphone, sampled at 4.681 kHz, was collected using a laptop with an RS232 connection to the SensorButton. This data was then labelled by hand - see Figure 5 for an example of the labelled data from one of the subjects.

The experiment produced a dataset of 8303 seconds. About 33% of the total time is taken up by *NULL*.

IV. Recognition methods

Recognition is carried out by combining the output of two classifiers (sound and acceleration) over a fixed-width sliding window of length w_{len}. For sound

Fig. 4. The kitchen activities used in the experiment.

a spectrum pattern matching method based on Linear Discriminant Analysis (LDA) is used. For acceleration, Naive Bayes (NB) is used. The methods used here are similar to those of our earlier work on carpentry tool-use recognition [17].

A. Sound recognition using LDA

The LDA classification is carried out on a short sliding window (or frame) of data, w_{lda}. In keeping with previous work, we use w_{lda} =100ms, which slides forward by 25ms after each calculation. At each step, an FFT is applied producing an M-dimension vector (with M typically being large). This vector is then reduced in dimension after multiplication by the LDA transform matrix (obtained from training data). The resulting $N(= \#Classes - 1)$ dimension vector is then compared to its class means (also obtained from training) to produce a list of class distances.

To produce a single result for the larger window w_{len}, the constituent LDA distance vectors must be combined. This is done by taking the mean of the LDA distance vectors over w_{len} for each class. Classification is then simply a matter of choosing the minimum mean distance vector.

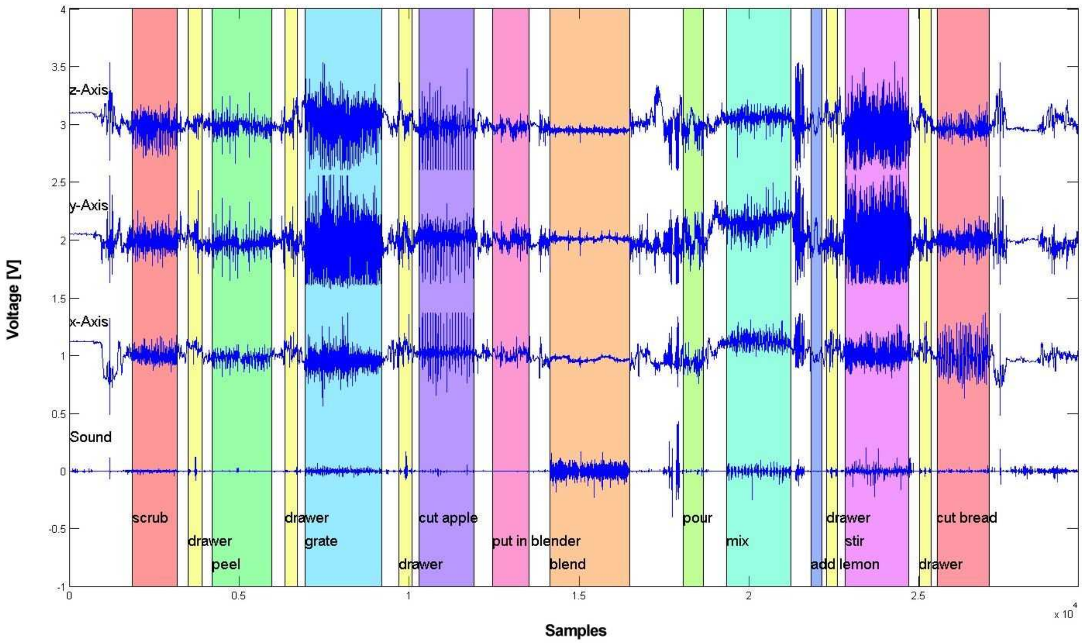

Fig. 5. Acceleration and sound signals acquired from one subject. Shaded areas mark hand-labelled ground truth for each activity.

1. scrub the carrot
2. take the peeler from the drawer
3. peel the carrot
4. take the grater from the drawer
5. grate the carrot
6. take the knife from the drawer
7. cut the apple into slices
8. put everything into the blender
9. blend everything
10. pour the blended carrots and apples into a bowl
11. mix it further with a hand mixer
12. add some lemon juice
13. take the fork from the drawer
14. stir the juice with the fork
15. cut 3 slices of bread

TABLE II

RECIPE FOR APPLE-CARROT JUICE

B. Acceleration recognition using NB

From the 109 Hz sampled raw acceleration data, two different feature types are extracted for each of the raw x, y & z-axes signals: the mean value over w_{len}; and a count on the number of signal peaks over w_{len}. The mean features help to give an estimation of the position and movement of the hand, whereas the peak count features help give a rough estimate of frequency.

The pre-calculated probability densitiy function (pdf), obtained from a training step, can then be used to approximate the NB likelihood for each class given the incoming features.[1]

C. Comparison of top results (COMP)

Because the classifiers are based on completely different sensing modalities - sound and acceleration - the chance that they will agree on a false classification is low. In fact, given 12 classes, the probability of such an occurrence happening randomly is $12^{-2} = 0.007$, or about 0.7%.

COMP simply compares the two classifier outputs and returns only those results which agree. If the classifiers do not agree, *NULL* is returned.

V. PRELIMINARY RESULTS (USING ALL 12 ACTIVITY CLASSES)

The LDA and NB methods all require training of parameters using data. This was carried out in a user-dependent, leave-one-out fashion. This is where, for each user, one set is put aside for testing while the remaining sets (from the same user) are used for training.

The system was initially evaluated across a sweep of the window length parameter w_{len}. In an early study (using just five of the eight subjects) this was found to have most effect on the NB based classifier, as can be seen by the varying recognition rates for the different

[1]Rather than calculate full probabilities, only the likelihoods from the NB calculation are used in classification.

classes in Figure 6. Setting w_{len} to 2 seconds was found to produce a suitable compromise. Intuitively, the suitability of such a large window stems from the fact that all activities of interest in these experiments occur at a timescale of at least several seconds. All further analysis was carried out with this parameter set.

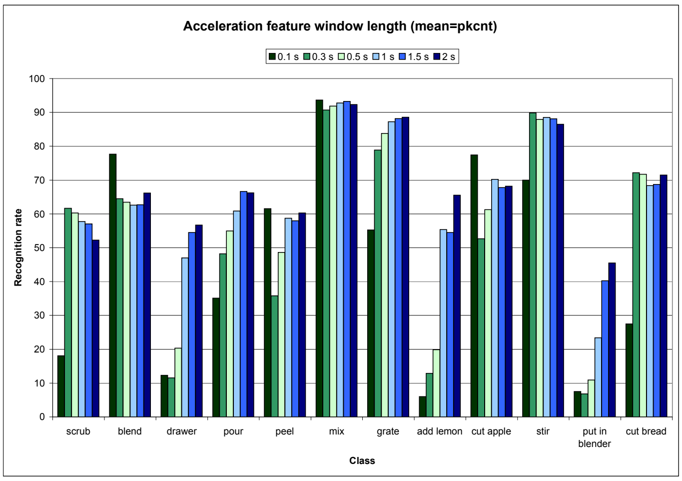

Fig. 6. Influence on average class recognition (class relative recall) for a selection of acceleration feature windows. The same size of sliding window is used for both mean and peak count features.

For the FFT, 16, 32, 64, 128 and 256-bit implementations were evaluated. The average results of this for each class are shown in Figure 7. For many classes, a full 256-bit FFT produced the best results. However the 32-bit FFT was used in this work so as to avoid excessive computation overheads on the (16-bit) MSP.

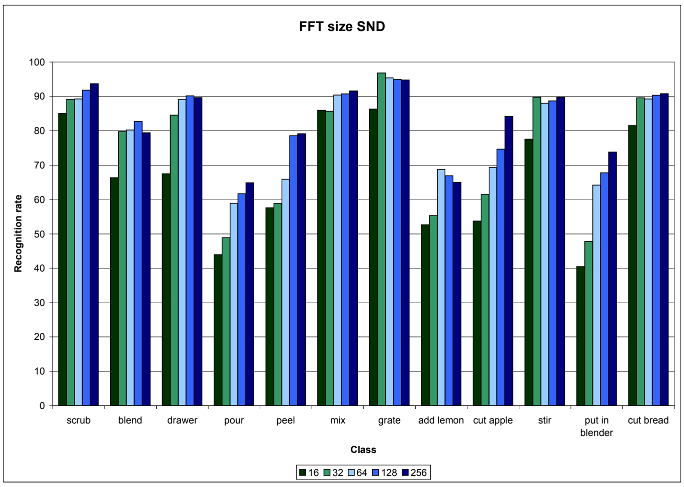

Fig. 7. Influence on average class recognition rates for a selection of FFT bit resolutions.

From the preliminary studies shown in Figure 6 and Figure 7, it can be seen that some classes perform worse than others. When LDA and NB classifiers are combined, the classes peel, drawer, 'put in blender' and 'add lemon' become particularly troublesome. This is largely due to the often very undefined manner in which these actions are carried out: drawer, for example, involved subjects opening and closing the drawer in a variety of ways - sometimes not even using their hands. For this reason, these classes were removed from further analysis by assigning their labelling to NULL.

For the remainder of the results presented here, a subset of eight activity classes were used. As a result of relegating the poorly performing classes, the percentage of *NULL* in the dataset grew from around 33% to 53%.

VI. Results (using eight activity classes)

One method of evaluating continuous recognition results is to compare ground truth and prediction sequencies in a timewise (frame by frame) performance analysis. Information on Correct Positives (CP), True Negatives (TN), False Positives (FP), False Negatives (FN) and Substitution errors, can then be summarised.

Figure 9 shows a bar chart of the basic summarising counts for CP, TN, FP, FN and Substitutions over the entire time of the eight-subject dataset. The LDA, NB, and their combination results are shown here, all of which are obtained from the Matlab analysis. Notice how the combination method is able to return *NULL*, i.e. TN and FN, where the constituent classifiers cannot. The combination method is able to drastically reduce substitution errors from 6.8% for NB and 4.6% for LDA, down to 0.3% (of the total time).

To get a more complete picture of these results, these representations can be extended to take account of additional information that is not available using traditional methods. In continuous time-based recognition, the edges of events are often ill-defined and can be subject to *overfill* and *underfill*. These are designated as follows:

- overfill: when a continuous sequence of correct prediction frames slip over the ground truth boundary to cover *NULL*;
- underfill: the time left when a continuous sequence of correct prediction frames does not completely cover the corresponding ground truth.

Additionally, continuous recognition results representing a single event might be broken up into several; or several events might be merged into one recognised event. Two designations of error which account for these phenomena - common to activity recognition problems - are *fragmenting* and *merge*:

- fragmenting: when a continuous sequence of correct prediction frames have been 'broken up' by small insertions of *NULL*
- merge: a single long continuous prediction covers two or more separate ground truth events.

Figure 8 gives an idea of how these errors might look (for a two-class output). For a more detailed account of this method see [18].

The introduction of these additional categories means that FP and FN are now subdivided into six: FP is divided into overfill, merge and insertion; FN is divided into underfill, fragmenting and deletion. The Serious Error Level (SEL) is defined as the percentage of total experiment time involving merge, insertion, fragmenting, deletion and substitution errors.

In Figure 10 we see that underfill errors take up 4.7% of the total error time (previously classed as FN); and overfill takes up about 3% (previously FP). [2] Although the overall error rate of our system is about 21%, the serious error rate is much less (14%).

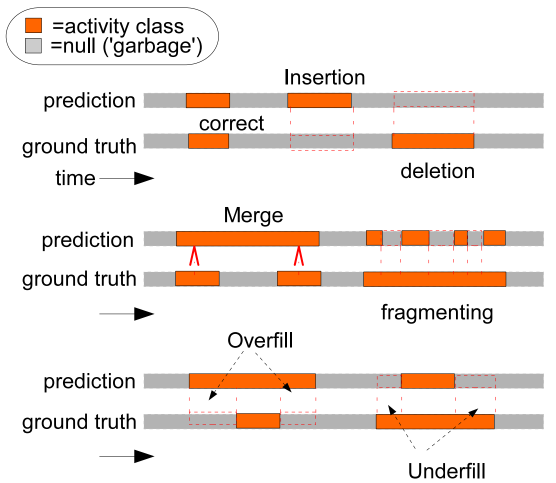

Fig. 8. Different types of error in a timewise continuous recognition evaluation: insertion, deletion, merge, fragmenting, overfill and underfill.

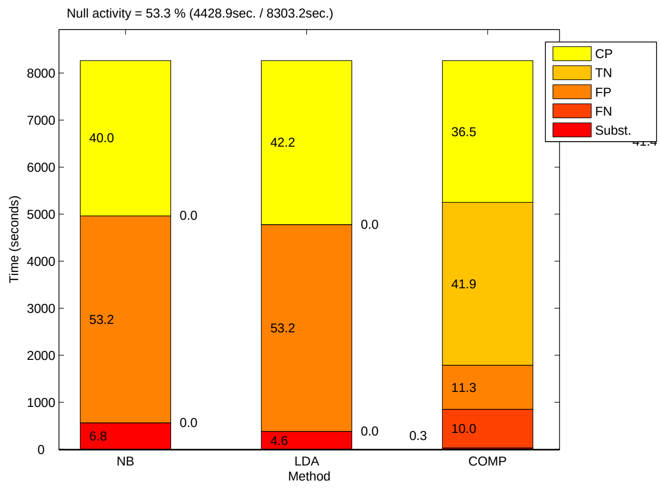

Fig. 9. A breakdown of errors as a percentage of total experiment time for NB, LDA and COMP: Correct Positive (CP), True Negative (TN), False Positive (FP), False Negative (FN) and Substitution times.

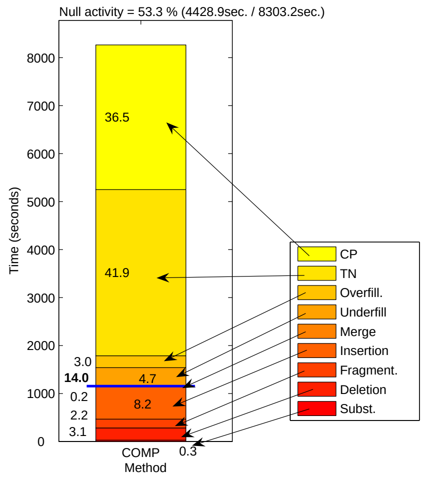

Fig. 10. Error Division Diagram (EDD): a breakdown of errors as a percentage of total experiment time for COMP method: Correct Positive, Correct Negative, Overfill, Underfill, Merge, Insertion, Fragmenting, Deletion and Substitution times; also given is the 'serious error' level, which ignores the minor errors of Overfill and Underfill

A. Results running on IAR-workbench MSP simulation

We implemented the algorithms in C and ran them on the IAR- workbench simulation of the MSP used in SensorButton. Using the pre-recorded dataset, we performed the same analysis as was carried out in Matlab. The results from this are shown in Figure 11

These results show a drop in overall performance by about 30% of the total time - which translates to 7% if we consider only serious error (SEL) - compared to the Matlab implementation. This degradation in performance manifests itself by the large increase in substitution errors for both NB and LDA. One reason for this is the necessary reductions in resolution required to implement floating point calculations for both FFT and NB on the MSP - a processor which does not naturally support such operations.

[2]Note also that fragmenting errors occur for 2.2% of the activity time (previously FN).

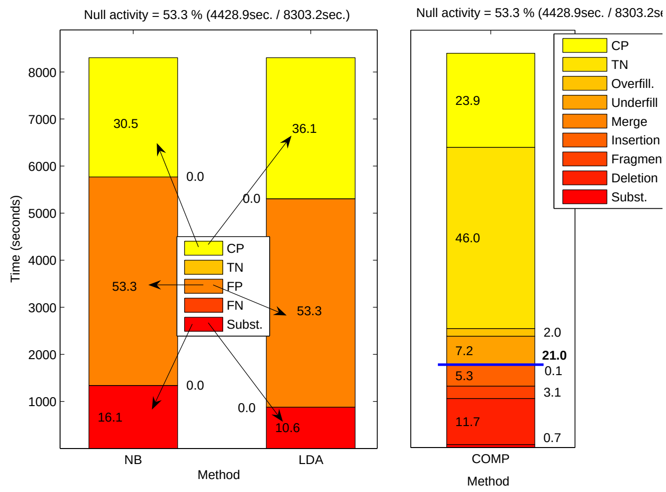

Fig. 11. IAR on-device simulation: NB and LDA results (left); and EDD for COMP (right)

B. On-device operation

All the algorithms - data collection, feature calculation, continuous recognition - were compiled and run on the SensorButton platform. The recognition output, a single ASCII character for each of the LDA, NB and COMP algorithms, was transmitted back to a PC via wireless.

Only a preliminary function test could be carried out for the on-device recognition. In order to carry out a complete online, multi-subject study involving all aspects of the continuous recognition process, a new sequence of experiments must first be carried out[3].

From these intial tests however, the power consumption of the system during continuous recognition, including wireless transmission, was measured to be 22.22 mW. Using a lithium-poly rechargeable battery with 150 mAh current capacity it should be possible to power the system in full operation for around 25.25 hours.

VII. Conclusion

In this paper we showed how a wrist-worn sensor node, using a microphone and 3-axis accelerometer, can be used to detect a set of activities from a kitchen scenario. We have shown results in an offline multi-subject study that indicates an overall error rate of 20%. When overfill and underfill are accounted for, the frame by frame error drops to about 14% of the total experiment time. When running the algorithms, ported to 16-bit MSP430 code, on an IAR-Workbench simulation, the serious error goes up to around 21% (with overall error 30%).

[3]At time of writing, this was work in progress.

At full continuous operation the device consumes approximately 22mW. This means that with all the algorithms running, including the wireless transmission of the recognition results, the device can run unhindered for about 25 hours.

A. Future work

We plan to have a complete multi-subject study for the online device recognition. Further we plan to implement improvements to the recognition algorithms (e.g. using a more advanced classifier fusion).

We also hope to adapt a version of Tiny OS for use on the SensorButton. This should provide a more easy-to-use software platform for implementing different recognition algorithms. In addition we are implementing a multi-hop protocol for distributed on-body networking and classification using several sensor button nodes. This should provide a system that can perform activity recognition using joint information from different parts of the body.

a) Acknowledgments: We would like to thank Claudio Fomeli and Marco Graf for their contributions to this project. Thanks also to Matthias Staeger for his advice on the FFT implementation, and to Oliver Amft for his ground truth labelling software.

References

[1] D. Abowd, A. K. Dey, R. Orr, and J. Brotherton. Context-awareness in wearable and ubiquitous computing. *Virtual Reality*, 3(3):200–211, 1998.

[2] Krause Andreas, Matthias Ihmig, Edward Rankin, Smriti Gupta, Daniel Siewiorek, Asim Smailagic, Michael Deisher, and Uttam Sengupta. Trading off prediction accuracy and power consumption for context-aware wearable computing. In *In 9th IEEE Int'l Symp. on Wearable Computers*, pages 20–26, October 2005.

[3] S. Antifakos, F. Michahelles, and B. Schiele. Proactive instructions for furniture assembly. In *4th Int'l Conf. UbiComp*, page 351, Gteborg, Sweden, 2002.

[4] L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive, LNCS 3001*, 2004.

[5] Nagendra B. Bharatula, Stijn Ossevoort, Paul Lukowicz, and Gerhard Tröster. Reliability modeling of embedded system-in-a-package: Design and packaging issues. In *The 2004 International conference on ESA/VLSI, Lasvegas, USA*, pages 387–392, June 2004.

[6] Nagendra B. Bharatula, Mathias Stäger, Paul Lukowicz, and Gerhard Tröster. Power and size optimized multi-sensor context recognition platform. In *In ' 9 th IEEE International Symposium on Wearable Computers, ISWC 05', Osaka, Japan*, pages 195–196, October 2005.

[7] Nagendra B. Bharatula, Rolf Zinniker, and Gerhard Tröster. Hybrid micropower supply for wearable pervasive sensor nodes. In *In ' 9 th IEEE International Symposium on Wearable Computers, ISWC 05', Osaka, Japan*, pages 197–198, October 2005.

[8] M.C. Büchler. *Algorithms for Sound Classification in Hearing Instruments*. PhD thesis, ETH Zurich, 2002.

[9] Graeme S. Chambers, Svetha Venkatesh, Geoff A. W. West, and Hung H. Bui. Hierarchical recognition of intentional human gestures for sports video annotation. In *Int'l Conf. on Pattern Recognition*, 2002.

[10] Jianfeng Chen, Alvin Harvey Kam, Jianmin Zhang, Ning Liu, and Louis Shue. Bathroom activity monitoring based on sound. In *Pervasive, LNCS*, volume 3468, pages 47–61, 2005.

[11] G. Fang, W. Gao, and D. Zhao. Large vocabulary sign language recognition based on hierarchical decision trees. In *Intl. Conf. on Multimodal Interfaces*, Vancouver, BC, Nov. 2003.

[12] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa. Computational auditory scene recognition. In *Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, volume 2, pages 1941–1944, May 2002.

[13] P.Lukowicz, J.A.Ward, H. Junker, G. Tröster, A.Atrash, and T.Starner. Recognizing workshop activity using body worn microphones and accelerometers. In *Pervasive, LNCS*, 2004.

[14] James Scott and Boris Dragovic. Audio location: Accurate low-cost location sensing. In *Pervasive, LNCS*, volume 3468, pages 1–18, 2005.

[15] M. Stäger, P. Lukowicz, and G. Tröster. Implementation and evaluation of a low-power sound-based user activity recognition system. In *Proc. IEEE Int'l Symp. on Wearable Comp.*, 2004.

[16] T. Starner, B. Schiele, and A. Pentland. Visual contextual awareness in wearable computing. In *Proc. IEEE Int'l Symp. on Wearable Comp.*, pages 50–57, Pittsburgh, PA, 1998.

[17] J.A. Ward, P. Lukowicz, and G. Tröster. Gesture spotting using wrist worn microphone and 3-axis accelerometer. In *Soc-Eusai '05, Proceedings*, Oct. 12-14 2005.

[18] J.A. Ward, P. Lukowicz, and G. Tröster. Evaluating performance in continuous context recognition using event-driven error characterisation. In Mike Hazas, John Krumm, and Thomas Strang, editors, *LoCA*, volume 3987 of *Lecture Notes in Computer Science*, pages 239–255. Springer, 2006.

[19] J.A. Ward, P. Lukowicz, G. Tröster, and T. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(10):1553–1567, October 2006.

[20] Tracy Westeyn, Kristin Vadas, Xuehai Bian, Thad Starner, and Gregory D. Abowd. Recognizing mimicked autistic self-stimulatory behaviors using hmms. In *Proc. IEEE Int'l Symp. on Wearable Comp.*, pages 164–169, 2005.