Activity Recognition of Assembly Tasks Using Body-Worn Microphones and Accelerometers

Jamie A. Ward, *Member, IEEE,* Paul Lukowicz, *Member, IEEE,*
Gerhard Tröster, *Member, IEEE,* Thad Starner, *Member, IEEE*

Manuscript received May 24, 2005; revised December 22, 2005

J. A. Ward and G. Tröster are with the Swiss Federal Institute of Technology (ETH), Institute for Electronics, Zürich, Switzerland. E-mail: {ward, troester}@ife.ee.ethz.ch

P. Lukowicz is with the Dep. of Computer Science, University of Passau, 94030 Passau, Germany. E-mail: paul.lukowicz@umit.at

T. Starner is with the Georgia Institute of Technology, College of Computing, 85 5th st. NW. TSRB, Atlanta GA 30332, USA. E-mail: thad@cc.gatech.edu

Abstract

In order to provide relevant information to mobile users, such as workers engaging in the manual tasks of maintenance and assembly, a wearable computer requires information about the user's specific activities. This work focuses on the recognition of activities that are characterized by a hand motion and an accompanying sound. Suitable activities can be found in assembly and maintenance work. Here, we provide an initial exploration into the problem domain of continuous activity recognition using on-body sensing. We use a mock "wood workshop" assembly task to ground our investigation.

We describe a method for the continuous recognition of activities (sawing, hammering, filing, drilling, grinding, sanding, opening a drawer, tightening a vise, and turning a screwdriver) using microphones and 3-axis accelerometers mounted at two positions on the user's arms. Potentially "interesting" activities are segmented from continuous streams of data using an analysis of the sound intensity detected at the two different locations. Activity classification is then performed on these detected segments using linear discriminant analysis (LDA) on the sound channel and hidden Markov models (HMMs) on the acceleration data. Four different methods at classifier fusion are compared for improving these classifications. Using user-dependent training, we obtain continuous average recall and precision rates (for positive activities) of 78% and 74%, respectively. Using user-independent training (leave-one-out across five users), we obtain recall rates of 66% and precision rates of 63%. In isolation, these activities were recognized with accuracies of 98%, 87%, and 95% for the user-dependent, user-independent, and user-adapted cases, respectively.

Index Terms

Pervasive computing, Wearable computers and body area networks, Classifier evaluation, Industry

I. Introduction

For office workers, computers have become a primary tool, allowing workers to access the information they need to perform their jobs. For more mobile workers such as those in maintenance or assembly, accessing information relevant to their jobs is more difficult. Manuals, schematics, system status, and updated instructions may be readily available on-line via wireless networks. However, with current technology, the user must focus both physically and mentally on a computing device either on his person or in the environment. For example, to access a specific schematic through a PDA, an aircraft repair technician needs to interrupt his work, retrieve his PDA from a pocket or bag, navigate the PDA's interface, read the desired information, and finally stow the PDA before resuming work. Equipping the worker with a head-up display and speech input or a one-handed keyboard, helps reduce distraction from the physical task. However, the worker's task is still interrupted, and he must make a cognitive effort to retrieve the required information.

For over a decade, augmented reality and wearable/ubiquitous computing researchers have suggested that pro-active systems might reduce this cognitive effort by automatically retrieving the right information based on user activity [1]. For example, as an airplane mechanic begins removal of a turbine blade from an engine, the manual page showing this procedure is presented automatically on his head-mounted display. The assumption is that such systems will be able to follow the progress of the task and automatically recognize which procedure is being performed. While other methods [2] are being explored, in this paper we assume such a continuous activity recognition system will use on-body sensors and computation to provide this facility.

A. Problem Analysis

We wish to explore the use of on-body sensors to recognize a user's activities. To ground our work, we have chosen to examine the activities involved in an assembly task in a wood workshop. For this exploration, we will focus on recognizing the use of five hand tools (hammer, saw, sanding paper, file, and screwdriver), the use of three machine tools (grinder, drill, and vise), and the use of two different types of drawers (which will be modeled in one class).

These activities, though limited here to a specific scenario, are fairly diverse. In some respects they can be said to provide insight into a wide range of activities using the hand and some object or tool. Common to many activities, they produce a broad range of different signatures for both sound and motion. Hammering, for example, is characterized by the rise and fall of the arm, accompanied on impact by a loud bang. Use of the saw produces a more regular sound, directly correlated with the back and forth movements of the arm. On the other scale, the hand twists associated with using a screwdriver are generally accompanied by correlated, quieter sounds. In contrast, the use of a drilling machine produces a loud, continuous sound, and whereas the motion of the arm during its use is also well-defined, it is usually independent from the sound being made. Even more extreme, the opening and closing of a drawer produces characterstic but widely varying sounds, with motions that can vary from a well-defined push and pull, to a simple nudge of the elbow or leg.

B. Paper Scope and Contributions

From these observations, microphones (sound) and accelerometers (motion) were chosen as suitable on-body sensors. In this paper we present the use of these devices, worn at two locations on the wrist and upper arm, to detect continuous activities in an assembly scenario. Specifically, we present:

1) **Two-microphone signal segmentation:** Through an apparatus similar in concept to a noise-cancelling microphone, we demonstrate a method for assisting segmentation of activities from a continuous stream - particularly for those activities where a noise is made close to the user's hand.
2) **Recognition using sound and acceleration:** Separate classifications are performed using spectrum pattern matching on the sound and Hidden Markov Models (HMM) on the acceleration data. We then compare various ways of fusing these two classifications. Specifically, we use methods based on ranking fusion (Borda count, highest rank, and a method using logistic regression) and a simple top class comparison.

The methods are evaluated using a multi-subject dataset of the wood workshop scenario. User-dependent, user-independent, and user-adaptive cases are evaluated for both isolated and continuous recognition to assess robustness of the methods to changes in user.

C. Related Work

Many wearable systems explore context awareness and pro-active involvement as means of reducing the cognitive load on the user [3]. Key to this is the ability to recognize user activities.

To date, much of the work in this area relies on the use of computer vision [4], [5], [6], [7], [8]. Though powerful, vision can suffer in the mobile and wearable domains from drawbacks such as occlusion and changes in lighting conditions as users move around. For many recognition tasks the computation complexity is often beyond what current wearable hardware can support.

Non-visual, body fixed sensors (BFS), in particular accelerometers, have been employed for many years in the analysis of body posture and activity [9], usually in a clinical setting [10], [11]. Using two uniaxial accelerometers - one radial at the chest, the other tangential at the thigh - Veltink *et al.* [12] were able to evaluate the feasibility of distinguishing postures, such as standing, sitting and lying; they also attempted to distinguish these from the dynamic activities of walking, using stairs, and cycling. Similar approaches, all with the goal of ambulatory recognition, have since been investigated [13], [14].

Uiterwall *et al.* [15] performed a feasibility study on the long term monitoring of ambulatory activities in a working environment - specifically maintenance and messenger work. In the wearable domain these activities have been addressed by a number of researchers as part of a general attempt at recognizing context [16], [17], [18]. 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 [19], or embedded in special gloves [20].

The use of sound has been investigated by Pelton *et al.* [21] for their work in analysing user situation. Intelligent hearing aids have also exploited sound analysis to improve their performance [22]. In the wearable domain Clarkson and Pentland used a combination of audio and video to infer situation based on short-term events (such as opening/closing doors) [23]. Wu and Siegel [24] used a combination of accelerometers and microphones to provide information about defects in material surfaces. For recognition of activities however, this combination of sensors has not been investigated to date.

Fusion of multiple information sources is a well-studied and diverse field covering many different disciplines. Within the domain of activity recognition, fusion of multiple sensors stems largely from the intuition that two well-placed sensors relay more information about an activity than one sensor alone. Combining the results from different classifiers has been investigated by numerous researchers [25], [26], [27]. The simplest method is to compare the top decisions of each classifier, throwing out any results in disagreement. The problem with this technique is that it disregards any particular advantage one classifier might have over another. Several alternative methods, all making use of class rankings, were explored by Ho *et al.* [28]. We apply these methods in this work to the specific problem of fusing sound and acceleration classifiers.

II. Recognition Method

To provide pro-active assistance for assembly and maintenance personnel, the computer needs to identify relevant activities from a continuous data stream. It has been shown that activity recognition in the isolation case - where the beginning and ending of activities are known - can be achieved with good accuracy [29]. However, in the continuous case where the start and completion of activities are not known, reliable recognition is still an open problem. The main difficulty lies in the fact that large segments of random, non-relevant activities often occur between activities meaningful to the task. These non-relevant activities can involve many diverse movements such as scratching one's head, swinging the arms, or taking something out of the pocket. This diversity means that it is infeasible to define a "garbage class" for the accelerometer data that is sufficiently well separated from the relevant activities.

We solve this problem by using sound analysis to identify relevant signal segments. Our approach is based on the assumption that all of the activities in which we are interested produce some kind of noise close to the hand. While this is certainly not true for many human activities, in our case it is a reasonable assumption as most assembly tools and machines make characteristic noises when in use. We thus define the null class by the absence of such a characteristic sound in the *proximity of the user's hand*. To this end we use the intensity difference between the microphones mounted on the wrist and upper arm. Further improvement of the segmentation is achieved through clustering of short frame-based sound classifications over longer sliding windows. We then treat those segments as isolated events on which both sound and acceleration classification is performed separately. Finally these separate classifications are fused. This step is particularly important for removing false positives resulting from the over sensitivity of the sound segmentation. Four different methods of fusion are evaluated: comparison of top choices

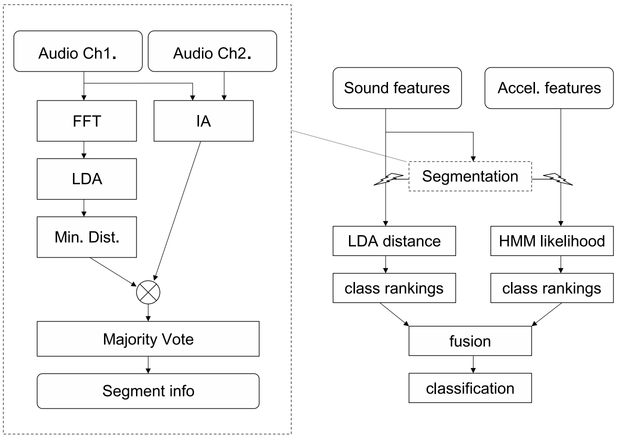


Fig. 1. Recognition algorithm: segmentation using two channels (wrist and arm) of sound (left); overall recognition process (right).

(COMP), highest rank, Borda count, and a method using logistic regression. An overview of the recognition process is given in Figure 1. Key steps are elaborated below.

A. Sound Intensity Analysis (IA)

Partitioning cues are obtained from an analysis of the difference in sound intensity from two different microphone positions [30]. This is based on the premise that most workshop activities are likely to be associated with a characteristic sound originating near the hand.

Since the intensity of a sound signal is inversely proportional to the square of the distance from its source, two microphones - *(1)* on the wrist, and *(2)* on the upper arm - will register two signal intensities (I_1 and I_2) whose ratio I_1/I_2 depends on the absolute distance of the source from the user. Assuming that the sound source is located at distance d from the first microphone and $d + \delta$ from the second, the ratio of the intensities is proportional to:

$$\frac{I_1}{I_2} \simeq \frac{(d+\delta)^2}{d^2} = \frac{d^2 + d\delta + \delta^2}{d^2} = 1 + \frac{\delta}{d} + \frac{\delta^2}{d^2}$$

Sound originating far from the user, $d >> \delta$, will result in $\frac{I_1}{I_2} \simeq 1$. Whereas sound originating close to the user's hand, $d \simeq \delta$, will result in $\frac{I_1}{I_2} > 1$. Thus, the ratio $\frac{I_1}{I_2}$ provides an indicator of whether a sound was generated from the action of the user's hand. Based on this, the following sliding window algorithm is performed over data from the two audio channels:

1) Slide window w_{ia}, in increments of j_{ia} , over both channels, calculating I_1 and I_2 at each step
2) For each frame, calculate $I_1/I_2 - I_2/I_1$: zero indicating a far off (or exactly equidistant) sound, while a positive value indicating a sound closer to the wrist microphone (1)
3) Select those frames where this ratio difference passes a suitable threshold T_{ia}

B. Frame-by-Frame Sound Classification Using LDA

Frame-by-frame sound classification is performed using pattern matching of features extracted in the frequency domain. Each frame represents a window of w_f =100ms of raw audio data (sampled at f_s = 2kHz). From this a Fast Fourier Transform (FFT) is performed generating a 100 bin output vector $(1/2 * f_s * w_f = 1/2 * 2 * 100 = 100\ bins)$. The choice of these parameters is based on preliminary investigations into achieving suitable recognition performance while minimizing computation requirements.

Making use of the fact that the recognition problem requires a small number of classes, Linear Discriminant Analysis (LDA) [31] is applied to reduce the dimensionality of the FFT vectors from 100 to the number of classes ($\#classes$) minus one. Classification of each frame can then be performed by calculating the Euclidean distances from the incoming point in LDA space to the mean of each class (as obtained from training data). Minimum distance is then used to select the top class[1]. The savings in computation complexity by dimensionality reduction come at the comparatively minor cost of requiring us to compute and store a set of LDA class mean values.

C. Sound-Based Segmentation

The initial approach to segmentation was simply to apply the IA algorithm, with $w_{ia} = 100ms$ and $j_{ia} = 25ms$, across a sweep of different thresholds, highlighting those frames of interest for LDA classification and marking the rest as null. This tended to produce a somewhat fragmented

[1] Equally, a nearest neighbor approach might be used. However, this was not found to produce any significant improvement for the dataset used here.

result with wildly varying partition sizes. To combat this, two different methods of “smoothing” using variations of the majority vote were applied. In each of these, a window of just over one second was moved over the data in one second increments. This relatively large window was chosen to reflect the typical timescale of the activities of interest.

The first approach at smoothing was to run a two-class majority vote window directly over the output of the IA algorithm. This process has the effect that in any given window, the class with the most number of frames (either “interesting” or “null”), wins and takes all the frames within the window. In the (rare) event of a tie, the null class is assigned.

The second approach, and the one chosen for the remainder of the work, is to perform a majority vote over already classified frames, as shown in the left box of Figure 1. Firstly a preliminary frame-by-frame LDA classification is performed on those frames selected by IA; those not selected by IA are “classified” as null. Then a jumping majority vote is run over all of the frames. This process differs from the previous approach in that in order to “win” a window, a class has to have both more frames accounted to it than any other non-null class, and more than $1/\#classes$ of the total number of frames. If no positive class wins, null is assigned.

The results from all three of these approaches, and the reason for choosing multi-class majority vote, is explored further in the results section IV-C.1.

D. Sound classification

Segments are defined as a sequence of one or more contiguous non-null windows. Being non-null by definition, classification of a segment can be regarded in isolation and is simply a matter of taking a winner-takes-all vote of the constituent frame classifications.

When higher level information about a segment is required, such as the likelihood of each possible class, then the problem is not so straightforward. One approach is to build a histogram entry for each class over the frame-by-frame classifications, thus providing an estimate of class probability. However, this method throws out potentially useful information provided by the LDA frame-by-frame classification. Another approach, adopted in this work, is to take the LDA distance values for each class and calculate their mean over all the frames. This provides a single set of class distance values for each segment. These distances themselves might not be mathematically useful, but their rank is. How these are then used in classifier fusion is elaborated in the Recognition Method section II-G.

E. Acceleration features

The 3-axis accelerometer data streams x, y and z, from both wrist and arm mounted sensors, are sampled at 100Hz. (The x-axis on the wrist is defined by drawing a line across the back of the wrist between the joints where the two forearm bones connect to the hand. The x-axis on the shoulder can be described as parallel to the line connecting the bicep and tricep muscles through the arm.) A short sample sequence of this data (x, y, z for wrist, and x for arm) for the activities of sawing, putting the saw in a drawer, clamping some wood with a vise, and using the drill, is shown in Figure 2. The locations of the sensors are also shown in this figure.

Selection of features is a critical task for good recognition performance. Since a thorough analysis into the best possible features is beyond the scope of this work - we are more concerned with recognition improvements through classifier fusion - we select features based on a combination of intuition and empirical experience of what works well for this problem. Specifically, the features calculated are a count on the number of peaks within a 100ms sliding window, the mean amplitude of these peaks, and the raw x-axis data from the wrist and arm sensors.

These features reflect our intuition (and the analysis of previous researchers also using tri-axial accelerometers [32]) that three main components will affect the readings: gravity, motion initiated by the user, and impacts of the hand with objects. Higher frequency vibrations will be associated with this last component, and counting the number of peaks in a 100ms window is a computationally inexpensive way to capture this effect. For example, a large number of peaks may indicate the “ringing” in the hand caused by the impact of, say, striking a hammer or pushing a saw into wood.

A smaller number of peaks may be caused when the user initiates a motion. Intuitively, the force the user’s muscles apply to the hand will result in a smooth acceleration as compared to the jerk (and higher order components) associated with impact events. For example, the twist of the screwdriver results in peaks in acceleration as the user starts and stops the twist.

The orientation with respect to gravity is also reflected in our features. The mean height of peaks in a 100ms window is composed of both 1g acceleration due to gravity and any other shock caused by interaction with the object or motion by the user. Gravity is represented even more explicitly in the raw x-axis data recorded from the wrist and arm. For example, twists of the wrist will show a large effect as the x-axis becomes perpendicular with the floor.

This last example illustrates an interesting point. A twist of the wrist associated with the turn of a screwdriver has a large effect at the wrist but a much smaller effect at the upper arm. Similarly, vibrations from machine tools affect the wrist much more than they do the upper arm. Thus, the upper arm can provide lower frequency posture information while the wrist provides cues as to the interactions with objects.

F. Acceleration classification

In contrast to the approach used for sound recognition, we employ Hidden Markov Models (HMMs) for classification of the accelerometer features [33], [34]. The implementation of the HMM learning and inference routines was provided courtesy of Kevin P. Murphy's HMM Toolbox for Matlab [35]. To increase the computation speed of these algorithms, the features are further downsampled to 40Hz (this has negligible effect on eventual recognition rates). They are also globally standardized so as to avoid numerical complications with the learning algorithms.

The HMMs use a mixture of Gaussians for the observation probabilities. The number of mixtures and hidden states are individually tailored by hand for each class model. Classification is performed by choosing the model which produces the largest log likelihood given a stream of feature data from the test set.

With the exception of drilling, all of the class models operate over a short time frame (e.g. around 1 second). As it is unlikely that a user will change activity more than once in this time, the recognition system is insulated from changes to the ordering in which activities are performed.

G. Comparison of top choices (COMP)

The first approach at fusion is the simplest of all the methods employed here. The final decision labels from each of the sound and acceleration classifiers for a given segment are taken, compared, and returned as valid if they agree. Those segments where the classifiers disagree are classified as null (no activity).

H. Fusion using class rankings

There are cases where the correct class is not selected as the top choice by one classifier, but may be listed second. Such near misses would be ignored if only classifier decisions were considered. A more tolerant approach considers levels of confidence a classifier has for each possible class. However, when combining information from different types of classifiers, the measures may be inconsistent or incomparable with one other.

In this case we use measures based on LDA distance and HMM class likelihoods. It is conceivable that these measures might be converted into probabilities and then fused using some Bayesian method, but this approach would require additional training in order to perform such a conversion. Additionally, with the view to a future distributed wearable sensing system, such computations might be expensive - for both calculation and, when one considers possible expansion of the number of classes, communication bandwidth. A mid-range solution is to consider the class rankings. This approach can be computationally simple and can lend itself to modular system design in case additional classes or classifiers are added at a later stage.

We use confidence measures to assign a ranking to each candidate. A classifier issues a list of class rankings which is compared to the rankings from the other classifiers. A final decision is made based on this comparison. To ensure that a decision is possible, rankings must be given a strict linear ordering, with “1” being the highest, and the lowest equaling the number of classes.

From the acceleration HMMs, an ascending rank can be produced directly from the inverse log likelihood of each class model (e.g. the largest likelihood being assigned the highest rank). For sound, the approach is slightly different. First, the LDA class distances for each frame in the segment are calculated. The mean of these is then taken, and ranking is assigned according to the criteria of shortest distance. Where there is a tie between classes, the ranking can be assigned randomly or, as in our case, by reverting to prior class preferences.

Three different methods of fusion using class rankings are used: highest rank, Borda count, and logistic regression. The implementation of each of these methods is described below:

1) Highest rank (HR): For any given input, take the rankings assigned to each class by the classifiers and choose the highest value. For example, if the sound classifier assigns “drilling” with rank “2” and the acceleration classifier gives it rank “1”, the highest rank method will return rank “1.”

This method is particularly suited to cases where for each class there is at least one classifier that is capable of recognizing it with high accuracy. It is also suitable for systems with a small number of classifiers - more classifiers might produce too many ties between class rankings.

2) Borda count: The Borda count is a group consensus function - the mapping from a set of individual rankings to a combined ranking. It is a generalization of majority vote: for each class it is the sum of the number of classes ranked below it by each classifier. The output is taken from ranking the magnitude of these sums, e.g. highest Borda count is assigned the highest rank.

Borda count is simple to implement, but it retains the drawback of all fusion mechanisms mentioned so far in that it treats all classifiers equally. To address this shortcoming, a method based on logistic regression was employed to approximate weightings for each classifier combination.

3) Logistic regression (LR): If the Borda count was extended to include a weighting on each combination of classifier rankings for every class, the fusion problem would soon become prohibitively expensive to calculate - especially for a large number of classes. One way to address this is to use a linear function to estimate the likelihood of whether a class is correct or not for a given set of rankings. Such a regression function, estimating a binary outcome with $P(true|X, class)$ or $P(false|X, class)$, is far simpler to compute. For each class a function can be computed, $L(X) = \alpha + \sum_{i=1}^{m} \beta_i x_i$, where $X = [x_1, x_2, ..x_m]$ are the rankings of the class for each of the m classifiers, and α and β are the logistic regression coefficients. These coefficients can be computed by applying a suitable regression fit using the correctly classified ranking combinations in the training set.

To obtain the combined rank, $L(X)$ is estimated for each class given the input rankings. Classification is performed by choosing the class with maximum rank. This method allows the setting of a threshold on $L(X)$, thus enabling us to return a “null” classification if the combination seems extremely unlikely. This threshold is chosen empirically.

III. Experimental setup

Performing initial experiments on “real-world” live assembly or maintenance tasks is inadvisable due to the cost, safety concerns, and the ability to obtain repeatable measurements under experimental conditions. As a consequence we decided to focus on an “artificial” task performed

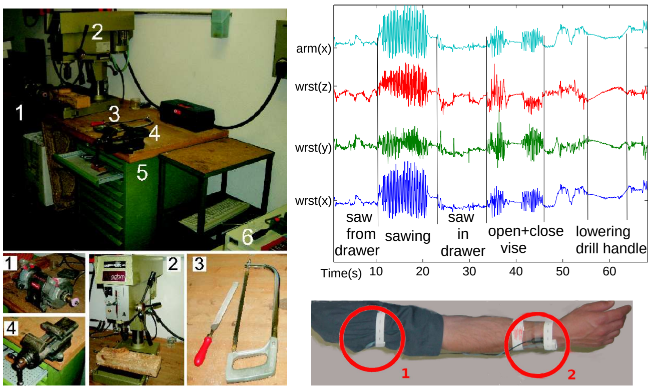


Fig. 2. The wood workshop (*left*) with *(1)* grinder, *(2)* drill, *(3)* file and saw, *(4)* vise, and *(5)* cabinet with drawers. Example of raw accelerometer data from the x-axis of arm, and x,y,z of wrist, for a subsequence involving saw, drawers, vise and drill (*top right*). Sensor placement (*bottom right*): *(1,2)* wrist and upper arm microphones and 3-axis acceleration sensors.

at the wood workshop of our lab (see Figure 2). The task consisted of assembling a simple object made of two pieces of wood and a piece of metal. The task required several processing steps using different tools; these were intermingled with actions typically exhibited in any real world assembly task, such as walking from one place to another or retrieving an item from a drawer.

A. Procedure

The exact sequence of actions is listed in Table I. The task was to recognize nine selected actions: use of hand tools such as hammer, saw, sanding paper, file and screwdriver; use of fixed machine tools such as grinder, drill and vise; and finally the use of two different types of drawer. To be ignored, or assigned as garbage class, are instances of the user moving between activities and of interactions with other people in the shop.

For practical reasons, the individual processing steps were only executed long enough to obtain an adequate sample of the activity. This policy did not require the complete execution of any one task (e.g. the wood was not completely sawn), allowing us to complete the experiment in a reasonable amount of time. However, this protocol influenced only the duration of each activity and not the manner in which it was performed.

Five subjects were employed (one female, four male), each performing the sequence in repetition between three and six times producing a total of (3+3+4+4+6)=20 recordings. Some

No	action
1	take the wood out of the drawer
2	put the wood into the vise
3	take out the saw
4	saw
5	put the saw into the drawer
6	take the wood out of the vise
7	drill
8	get the nail and the hammer
9	hammer
10	put away hammer, get driver and screw
11	drive the screw in
12	put away the driver
13	pick up the metal
14	grind
15	put away the metal, pick up wood
16	put the wood into the vise
17	take the file out of the drawer
18	file
19	put away the file, take the sandpaper
20	sand
21	take the wood out of the vise

TABLE I

STEPS OF WORKSHOP ASSEMBLY TASK.

subjects performed more repetitions than others because of a combination of technical problems in recording data and the availability of subjects. Each sequence lasted five minutes on average.

For each recording, the activity to be performed was prompted automatically by a computer, which an observer announced vocally to the subject. The exact timing of each activity was recorded by the computer when the observer pressed a key at the beginning and end of the activity. Any errors in these semi-automatic annotations were later corrected by visual inspection of the data and listening to the recorded audio. This provided the ground truth from which all subsequent training and evaluations were based.

The definitions of activity start and stop during ground truth annotation might be judged differently by different observers. Differences again arise depending on which sources are used (visual, sound, or even acceleration signals). As such no labelling scheme of a continuous system can be perfect. For these experiments therefore, a set of definitions was drawn up of which the main aim was to at least maintain consistency between the different recordings.

B. Data Collection System

Data collection was performed using the ETH PadNET sensor network [36] equipped with two 3-axis accelerometer nodes connected to a body-worn computer, and two Sony mono microphones connected to a MiniDisk recorder. The sensors were positioned on the dominant wrist and upper arm of each subject, with both an accelerometer node and microphone at each location, as shown in Figure 2. All test subjects were right handed. These recordings were later ported to a desktop PC for processing. The two channels of recorded sound, initially sampled at 48kHz, were downsampled to 2kHz for use by the sound processing algorithms.

Each PadNET sensor node consist of two modules. The main module incorporates a MSP430149 low power, 16-bit mixed signal microprocessor (MPU) from Texas Instruments running at a 6MHz maximum clock speed. The current module version reads a maximum of three analog sensor signals (including amplification and filtering) and handles the communication between modules through dedicated I/O pins. The sensors themselves are hosted on an even smaller "sensor-module" that can be either placed directly on the main module or connected through wires. In the experiment described in this paper sensor modules were based on a 3-axis accelerometer package consisting of two ADXL202E devices from Analog Devices. The analog signals from the sensor were lowpass filtered in hardware with a $f_{cutoff} = 50Hz$, 2nd-order, Sallen Key filter and digitized at 12-bit resolution using a sample rate of 100Hz. [2]

IV. Results

A. Leave-One-Out Evaluation

All training for LDA, HMM and LR is carried out using three variations of leave-one-out:

[2] With these settings some aliasing is possible, but was not found to affect the experiments described.

1) *User-dependent*, where one set is put aside for testing, and the remaining sets from the same subject used for training.
2) *User-independent*, where data from the subject under test is evaluated using training data provided by the other subjects. This is the most severe test - evaluating the system's response to a never-before seen subject.
3) *User-adapted*, where one set is put aside for testing, and all remaining sets from all subjects are used for training. This case emulates situations where the system is partially trained for the user.

These methods are applied consistently throughout the work, and results for each are given where appropriate.

B. Isolation Results

As an initial experiment, the positive (non-null) events specified by ground truth are evaluated in isolation. The metric used is *isolation accuracy* (also known as *class relative sensitivity*), defined as $\frac{correct_c}{total_c}$, with the number of $correct_c$ and $total_c$ positive events for each class c.

Table II shows results for (a) user-dependent, (b) user-independent, and (c) user-adapted. Being an isolation test, null is not defined; however in the case of COMP, there is the possibility that an event be declared null, i.e. a *deletion*. For COMP almost all errors are infact deletions, and so the substitutions, where occurring, are highlighted in brackets.

As shown in Table II(a), most classes with user-dependent training produce very strong results for sound and acceleration (above 90%, for non-vise and drawer activities). Any substitution errors that do exist are then completely removed when the classifier decisions are compared (COMP), albeit at the expense of introducing deletions. The ranking fusion methods fare even better - with Borda recognizing five classes perfectly, and four with only a single event error.

When applied to data from subjects not in the training set (user-independent Table II(b)), an expected drop in recognition rates can be seen for sound and acceleration. Activities such as using the drill or drawer continue to register almost perfect results though, largely due to the specific movements which they require and the correspondingly person-independent sounds which they produce. Some activities, such as driving a screw and using a vise, yield poor results

Class	Total	sound	accel.	COMP	HR	Borda	LR
hammer	20	100	100	100	100	100	100
sawing	20	100	90	90	90	95	100
filing	20	95	75	70	95	100	95
drill	20	100	100	100	100	100	95
sand	20	95	95	90	95	95	95
grind	20	100	90	90	100	100	100
screw	20	85	95	85	95	95	95
vise	160	87.5	99.4	87	99.4	100	99.4
drawer	440	98.2	99.1	98	99.3	99.3	99.3
Average%		95.6	93.7	90.1	97.1	98.3	97.6

(a) User-dependent isolation accuracies

Class	Total	sound	accel.	COMP	HR	Borda	LR
hammer	20	90	85	75	70	75	85
sawing	20	75	45	35	35	70	80
filing	20	25	25	10(10)	10	50	60
drill	20	100	100	100	95	100	95
sand	20	60	70	35(5)	60	80	75
grind	20	85	35	30(5)	90	90	95
screw	20	85	95	85	95	95	95
vise	160	79.4	96.9	78(1)	97.5	99.4	97.5
drawer	440	95	96.4	92.1	99.1	98.6	98.2
Average%		77.2	72	60	86.3	84.2	86.7

(b) User-independent isolation accuracies

Class	Total	sound	accel.	COMP	HR	Borda	LR
hammer	20	100	100	100	85	85	95
sawing	20	85	65	60	60	75	90
filing	20	60	70	35	50	90	85
drill	20	100	100	100	100	100	100
sand	20	60	100	60	90	90	95
grind	20	95	75	70	100	95	100
screw	20	90	95	90	95	95	95
vise	160	85.6	96.9	83.8	97.5	98.8	96.9
drawer	440	96.4	98.9	95.7	99.6	99.3	99.6
Average%		85.8	88.9	77.2	86.3	92.0	95.2

(c) User-adapted isolation accuracies

TABLE II

Isolation accuracies for sound, acceleration, and the four combination methods. Note: for COMP all errors are deletions (except where given in brackets).

from sound but are clearly recognizable in the accelerometer data. Again this is due to the unique person-independent motions which one must perform to use these tools.

With user-independent training, simple comparison of the classifier results fares less well. Although the number of substitution errors is low, the large discrepancy in performance of the constituent classifiers ensures that the possibility of agreement is almost as low as the possibility of disagreement. This effect causes a large number of deletions - particularly for filing, sawing, sanding and grinding. In contrast the ranking methods - particularly LR - resolve this problem extremely well. Of particular note is the case of filing: although 60% (12/20) accuracy is not ideal, it is an enormous improvement on the 25% of the constituent classifiers.

Finally, with the user-adapted test, Table II(c), the results improve again. For this, LR performs best - almost as well as with user-dependent.

C. Continuous Recognition Results

Defining appropriate evaluation metrics is difficult in continuous activity recognition research [37]. There is no application independent solution to this problem [38]. Often the continuous recognition task requires discrimination of relatively rare activities from a default “null” activity that constitutes the majority of the time in the data. In addition, there may be more than one type of error in a system, such as posed by multi-class continuous recognition, and the common metric of accuracy can be misleading [39]. Further problems arise when one wishes to evaluate continuous recognition with ill-defined, often fragmented and variable length class boundaries. Similar problems exist in vision, and though ways of automatically dealing with them exist, e.g. for 2D graphics [40], it is common for researchers simply to show typical output figures e.g. [41]. A typical output of our system is shown in Figure 3. Although these results can be compared (and evaluated) visually against the hand-labelled ground truth, for large datasets it is desirable to have some automatic metric.

1) Segmentation evaluation method: The purpose of this initial investigation is to evaluate, for each method, how well positive activities in a continuous stream are identified and segmented from null. There are four possible outcomes: those returning positive activities, *true positive (TP)* and *false positive (FP)*; and those returning null, *true negative (TN)* and *false negative (TN)*. As the continuous recognition methods are all aimed at detecting TP activities, and null is simply what remains, TN is regarded as less critical than other outcomes. This is a similar view to that in Information Retrieval (IR), where the evaluation focus is on the positive results that are returned - how many of these are correct, and what proportion of the total existing positives they make up - rather than the remaining (often more numerous) negative results. The metrics chosen therefore are those common to IR, namely *precision* (also known as *positive prediction*

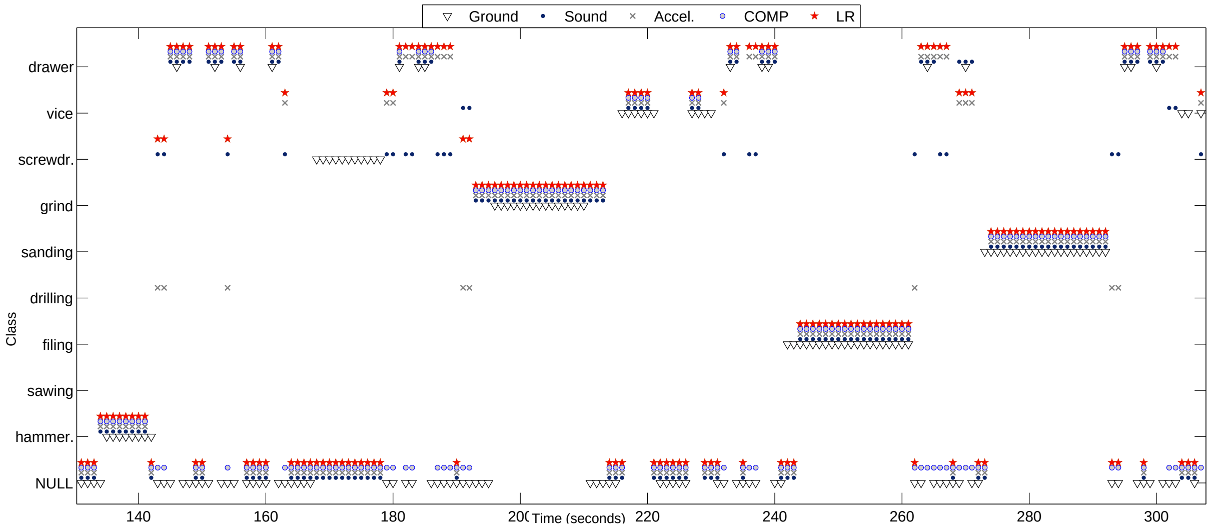


Fig. 3. Section of a typical output sequence (approx. 3 minutes). Ground truth is plotted alongside the sound and acceleration classifications, together with two approaches at fusing these - comparison (COMP) and logistic regression (LR). User-dependent training is used.

value) and *recall* (*sensitivity*, or *true positive rate*):

$$recall = \frac{true\ positive\ time}{total\ positive\ time} = \frac{TP}{TP+FN} \tag{1}$$

$$precision = \frac{true\ positive\ time}{hypothesized\ positive\ time} = \frac{TP}{TP+FP} \tag{2}$$

A *precision-recall* (PR) graph can be plotted to show the effects of different parameters when tuning a recognizer [42].

2) Segmentation results: In evaluating segmentation there are two parameters which can be varied: intensity analysis threshold T_{ia}, and the majority vote window size. Of these, T_{ia} has the most significant effect. For T_{ia} of (0, 0.1, 0.3, 0.5, 1, 1.5, 2, 3, and 5) the total, correct, and hypothesized times are calculated and summed over all test data sets. PR curves are then generated for each of the three segmentation schemes: IA selection on its own, IA smoothed with a majority vote, and IA+LDA smoothed with majority vote.

As expected, the IA alone gives the worst segmentation performance, with prediction output being heavily fragmented with false negatives and scattered with frames of false positive. The bottom curve in Figure $5(a)$ shows this performance across the range of thresholds. When a

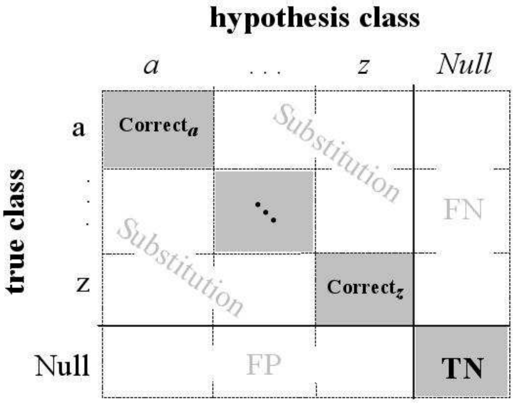


Fig. 4. Multi-class confusion matrix: diagonal marks the *correct positive* for positive classes, and True Negative (TN) for *NULL*; off-diagonal marks the positive class substitutions, the sum of the False Positives (FP) and the sum of False Negative (FN) errors.

majority vote is run over the IA selected frames however, many of the spurious fragmentation and inserted frames are smoothed away. Again this is reflected in the improved PR performance.

When we take the IA selected frames, apply LDA classification to them, and run a multi-class majority vote window over the entire sequence, the segmentation results are not immediately improved - in fact, for high precision, the IA+majority vote approach is still preferable. However, when considering that the later recognition stages will use fusion as a means of reducing insertions, a lower precision at the segmentation stage can be tolerated. With this in mind, high recall is preferable, and for this an improved performance can be seen using the IA+LDA+majority vote. A suitable recall rate of around 88% can be achieved with this method when the threshold of $T_{ia} = 0.3$ is chosen.

3) Continuous time (frame-by-frame) results: The sound and acceleration classifiers are applied to the partitioned segments. The four fusion algorithms are then applied on these.

Again PR curves are adopted, albeit with a slight modification to the precision and recall definitions so as to encapsulate the concept that in a multi-class recognition problem a TP data point is not just non-null, but can also be either a *correct* classification, or a *substitution*. Figure 4 gives a graphical breakdown of the possible designations as sections of a multi-class confusion matrix. The revised definitions of *correct recall* and *correct precision* are then given as:

$$correct\ recall = \frac{correct\ positive\ time}{total\ positive\ time} = \frac{correct}{TP + FN} \tag{3}$$

$$correct\ precision = \frac{correct\ positive\ time}{hypothesized\ positive\ time} = \frac{correct}{TP + FP} \tag{4}$$

These modified metrics are then calculated from the summed confusion matrices of all test datasets for each value of T_{ia}. Figure 5 shows the curves for the (b) user-dependent, (c) user-independent, and (d) user-adapted cases.

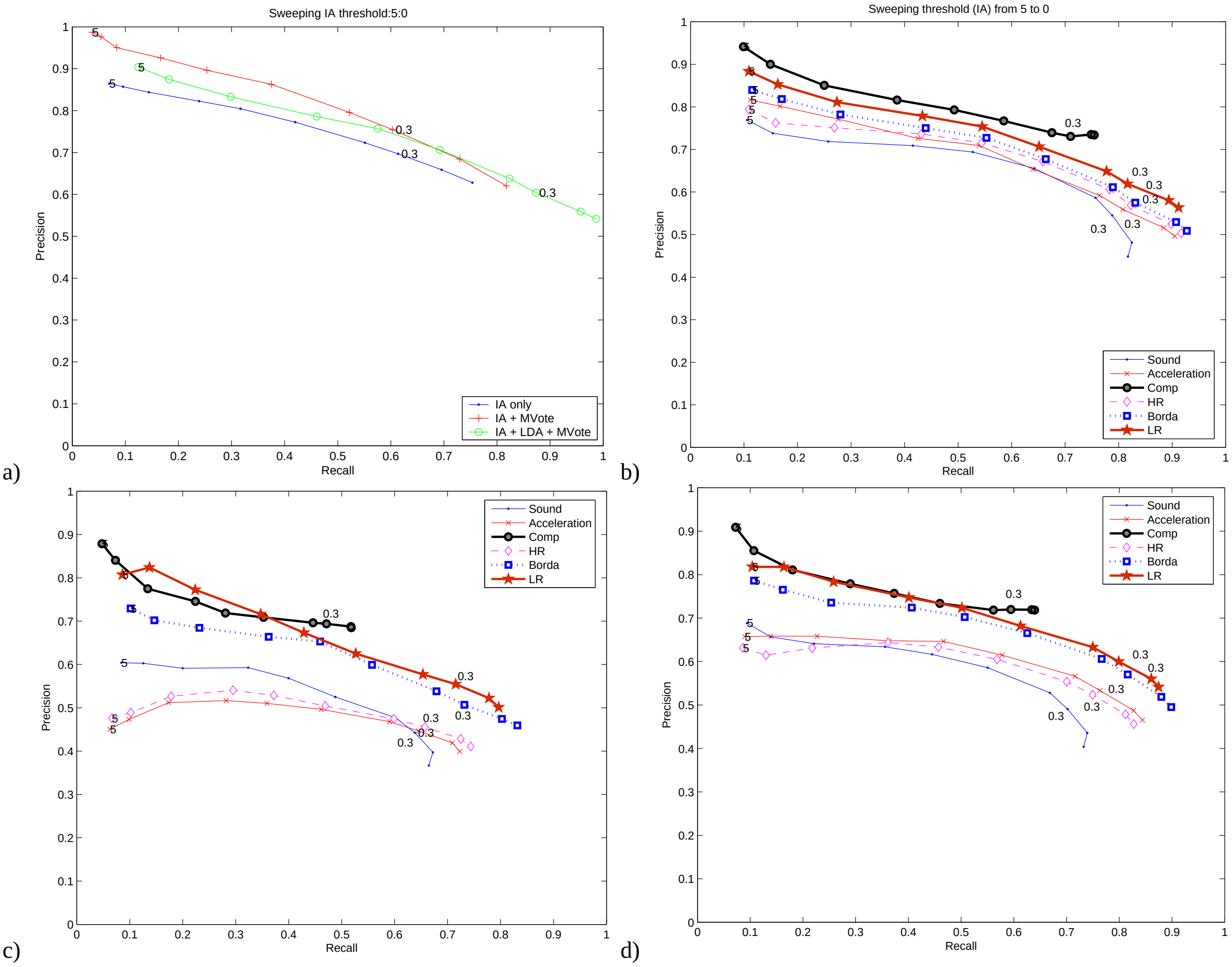


Fig. 5. Top left plot (a) shows PR comparison of 3 different segmentation schemes. The remaining plots show *correct PR* comparisons for the different classifiers and combination schemes, with user-dependent(b), independent(c) and adapted(d) cases.

a) Choosing a threshold: The main conclusion to be drawn from these graphs is that regardless of threshold, the classifiers and fusion methods perform relatively consistently with regard to each other within the precision-recall region of interest. LR always performs better than Borda, which performs better than HR, and this, in turn, is an improvement over the sound and accelerometer classifiers. Also noteworthy is the conclusion that $T_{ia} = 0.3$ yields consistency within a suitably close operating region for each of the methods, thus legitimizing further comparisons which require a fixed T_{ia}.

b) Confusion matrix based results: With T_{ia} set, the results can be examined in more detail. The first step is to calculate time-based confusion matrices, according to the template of Figure 4, and sum over all test datasets. Rather than present all twelve matrices (available on request from the authors), two summaries of the most pertinent results are made.

Firstly, the individual class performance is examined using *class relative* precision and recall. Recall is defined for each class, c as $\frac{correct_c}{total_c}$, and precision is defined as $\frac{correct_c}{hypothesized_c}$, where $correct_c$ is the total correct time, $total_c$ the total ground truth time, and $hypothesized_c$ the total time returned by the system, for class c. The precision and recall rates for each positive class, summarized by the averages over these, are shown in Table III. As an additional indicator of performance, *NULL* is included as a special class. Although the terms recall and precision are used for *NULL*, the recall of *NULL* is more accurately referred to as the system *specificity* $= \frac{TN}{TN+FP}$, with precision of *NULL* known as the *negative prediction value* (NPV) $= \frac{TN}{TN+FN}$.

Secondly, the overall performance, in terms of substitutions, FN, FP, TN and correct positive counts, is summarized in graphical form as the respective percentages of the total dataset size, as shown in Figure 6 (pending further discussion, only user-dependent is given).

4) Analysis of continuous frame-by-frame results: Based on the results of Table III the following observations can be made:

- Recognition performance is improved by fusion. Almost all classes improve over the constituent classifiers. One exception is with screwdriving, where performance is slightly lower than can be achieved by acceleration alone. An explanation for this result is the influence of extremely poor sound performance for this class.
- User independence. Recognition of machine tools, such as drill, grinder, vise and drawer is fairly user independent when using LR. With handheld tools, saw and hammer, there is a drop of roughly 10% in performance. Filing and sanding perform worst, almost certainly due to the greater variety of ways these activities can be performed.

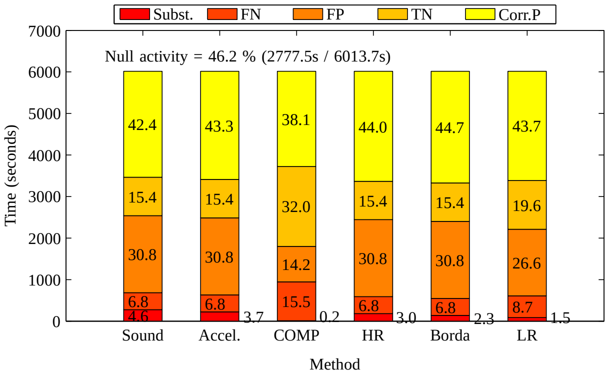


Fig. 6. Graphical summary of confusion matrix (user-dependent only): totals of the substitution, false negative (FN) and false positive (FP) error times are given as percentages of the total dataset time, together with true negative (TN) and correct positive times. Total count of *NULL* time in dataset is 46%.

- Performance of *NULL*. As the system has been tailored for recognition of positive events, it is not surprising that *NULL*, when treated as a class in its own right, performs poorly (e.g. 69/42 P/R for LR in (a)). COMP provides a compromise (e.g. P/R of 69/67 for (a)).

The summary in Figure 6 corroborates this first observation. Of particular note is the ability of the fusion methods to reduce substitution errors from approximately 3.7% of the total time in the acceleration classifier to as low as 1.5% for LR, and even 0.2% for COMP. The advantage of COMP is fewer false positives (FP) at the expense of more false negatives (FN). This is particularly evident when considering the very low recall rates of positive classes for this method in user-independent training, but COMP has the highest precision of all the methods. Correspondingly, it also has the highest recall of *NULL* (specificity) at 79%.

5) Event-based results: For many applications, frame-by-frame performance is of little significance. Of more interest is the detection of events that take place on a time scale of at least several seconds or hundreds of frames. For instance, when referring to “hammering,” we consider the whole consecutive hammering sequence contained in each experiment, not any individual hammer stroke. The corresponding definition of an event is a continuous time segment throughout which the system has returned the same classification. This definition can in principle be extended to segments of *NULL* as a “no activity event”.

Evaluation of event performance is similar to the strategy used in speech and character

	Sound		Accel.		COMP		LR	
Class (s)	%R	%P	%R	%P	%R	%P	%R	%P
hammer (196)	92	74	93	79	92	94	92	93
saw (306)	90	87	90	80	88	95	93	90
file (305)	77	80	80	82	65	94	82	90
drill (242)	95	54	99	41	95	64	96	59
sand (313)	82	67	87	92	77	93	83	94
grind (278)	83	69	63	66	62	80	75	73
screwd.(260)	52	20	53	87	51	86	53	81
vise (678)	65	55	74	49	61	69	73	53
drawer (659)	86	47	88	39	69	51	87	39
Pos.Average%	76	62	76	68	73	79	78	74
NULL(2778)	33	69	33	69	69	67	42	69

(a) User-dependent

	Sound		Accel.		COMP		LR	
Class (s)	%R	%P	%R	%P	%R	%P	%R	%P
hammer (196)	83	66	76	59	67	93	84	77
saw (306)	71	75	53	51	36	84	78	77
file (305)	29	46	19	39	7	34	23	46
drill (242)	91	47	99	28	92	62	93	62
sand (313)	48	35	51	66	31	89	50	67
grind (278)	72	57	26	45	19	74	82	66
screwd.(260)	46	14	50	86	48	86	50	79
vise (678)	55	54	71	38	47	79	71	62
drawer (659)	81	46	72	38	54	53	89	37
Pos.Average%	61	51	55	52	48	71	66	63
NULL(2778)	33	68	33	68	79	56	42	62

(b) User-independent

	Sound		Accel.		COMP		LR	
Class (s)	%R	%P	%R	%P	%R	%P	%R	%P
hammer (196)	85	62	92	81	85	94	91	83
saw (306)	78	79	61	81	49	97	85	88
file (305)	48	52	58	66	23	88	49	89
drill (242)	94	56	99	46	94	64	94	64
sand (313)	49	42	85	75	43	93	85	76
grind (278)	78	64	82	54	78	72	82	70
screwd.(260)	49	18	51	87	51	87	51	80
vise (678)	65	56	74	56	61	80	74	65
drawer (659)	84	47	89	38	68	52	92	37
Pos.Average%	67	55	73	65	63	79	75	72
NULL(2778)	35	69	35	69	72	61	43	68

(c) User-adapted

TABLE III

CONTINUOUS % RECALL(R) AND PRECISION(P) FOR EACH POSITIVE CLASS, AND THE AVERAGE OF THESE; ALSO GIVEN ARE THE R & P VALUES FOR *NULL* (T_{ia} = 0.3, S = TOTAL TIME IN SECONDS).

recognition. Importance is placed on the ordering of letters and words, rather than the specific time their components are uttered. Table IV presents event based results using the standard metrics of insertion and deletion. We reduce each evaluation to a two-class problem, i.e. one class against all others combined. Thus any predicted instance of a class that does not overlap with a same class event in the ground truth is marked as an insertion; and any ground truth instance of a class that has no corresponding prediction of that same class is marked as a deletion. By overlap, we mean some rough correlation of the output with the ground event.

6) Analysis of event-based results: Table IV helps to confirm many of the observations from the earlier frame-by-frame analysis. Across all user training cases, fusion drastically reduces the number of insertions for most positive classes. For the user-independent case, the low recall/high precision of COMP is confirmed with a high number of deletions - in worst case, filing with 17 deletions out of 20 events - but with few insertions. Again for fewer deletions, the LR method is a better choice.

7) Combined time and event-based evaluation: There is some information which Tables III and IV fail to capture. For example, the sanding activity in (a) has a recall of 83% (an error of 17% existing class time), yet produces only one deletion (1/20=5% of existing class events). Is this because the deleted event is longer than the others, or is it because the other sanding events do not completely cover their ground truth? The answer is generally a bit of both. In this case, most of the error lies with the later cause. Such mismatches in event timing constitute a considerable portion of the total frame by frame errors in the experiments described in this paper. We have also found them to be common in other similar work [43], [44], and we conclude our results presentation with a closer look at timing issues.

We first solidify the notion of timing errors through the concepts of Overfill and Underfill:

- Overfill (t) - FP frames forming part of a correct event which strayed over its segment borders.
- Underfill (t) - FN frames left when the correct event does not completely cover its borders

Examples of these situations are illustrated in Figure 7. We use the above definitions to recalculate the evaluation presented in Figure 6. This leads to some frames previously considered false positive to become Overfill. Similarly some FN frames are re-evaluated as Underfill. Note

	Sound		Accel.		COMP		LR	
Class (T)	I	D	I	D	I	D	I	D
hammer (20)	33	0	18	0	0	0	2	0
saw (20)	17	0	15	0	1	0	7	0
file (20)	20	0	17	1	2	1	8	0
drill (20)	40	0	83	0	2	0	8	0
sand (20)	62	1	2	1	0	2	0	1
grind (20)	13	0	24	5	2	6	9	2
screwd. (20)	293	4	8	3	8	4	14	3
vise(160)	131	18	168	15	38	35	146	15
drawer (440)	47	8	86	31	14	110	85	30
NULL (740)	33	299	33	299	35	86	41	242

(a) User-dependent

	Sound		Accel.		COMP		LR	
Class (T)	I	D	I	D	I	D	I	D
hammer (20)	46	0	44	2	0	3	20	1
saw (20)	32	0	25	7	4	11	15	0
file (20)	27	9	13	14	4	17	19	14
drill (20)	71	0	175	0	3	0	5	0
sand (20)	76	7	25	7	2	11	20	8
grind (20)	42	1	39	12	3	14	21	1
screwd. (20)	322	3	5	3	5	3	12	3
vise(160)	132	32	223	20	6	55	80	19
drawer (440)	57	22	105	90	13	169	123	16
NULL (740)	32	311	32	311	31	27	50	232

(b) User-independent

	Sound		Accel.		COMP		LR	
Class (T)	I	D	I	D	I	D	I	D
hammer (20)	59	0	17	0	0	1	15	0
saw (20)	26	0	7	6	0	8	7	1
file (20)	25	4	17	4	3	10	4	5
drill (20)	34	0	79	0	1	0	1	0
sand (20)	70	6	15	2	0	7	11	2
grind (20)	28	1	58	1	2	1	7	1
screwd. (20)	285	3	4	3	4	3	11	3
vise(160)	126	26	101	19	3	44	68	19
drawer (440)	51	15	93	22	17	111	121	8
NULL (740)	33	291	33	291	38	43	43	229

(c) User-adapted

TABLE IV

CLASS RELATIVE EVENT ERRORS FOR EACH CLASS: T =TOTAL, I =INSERTIONS, D =DELETIONS.

that substitution, correct positive, and true negative frame counts are not affected. Thus the recalculation essentially subdivides FP and FN into ‘timing error’ components, which have no influence on event recognition, and ’serious error’ components, which do.

Figure 8 shows the results of such a recalculation. Here *serious error level (SEL)* is denoted by a thick line. This graph includes substitution time in addition to the serious error components

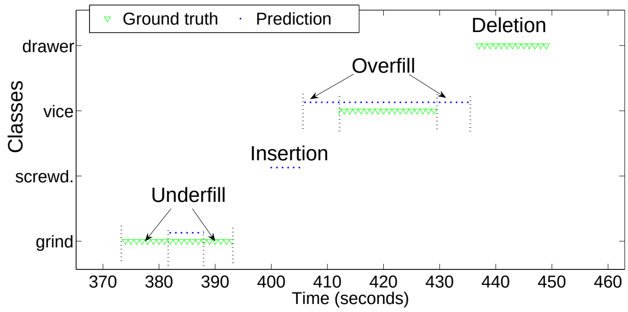


Fig. 7. Examples of Underfill, Insertion, Overfill, and Deletion errors.

of FP and FN. Errors below the serious error line would be considered part of an error for an event-based recognition system while errors above this line are timing errors and would be of more concern to a frame-by-frame recognition system. Thus the considerations presented in this paragraph can be considered as a combined time and event evaluation.

8) Analysis of combined time and event evaluation: The combined timing and event analysis provides a relatively complete characterization of system performance, from which the following observations can be made:

1) The correct positive time indicates the amount of time the correct activity was recognized, and the true negative time indicates the percentage of frames where the system correctly recognized that no activity was happening. These classes provide both an indication of the effectiveness of the recognizer as well as the difficulty of the problem. The sum of these two percentages indicate the standard frame-by-frame accuracy of the system. At a glance we see that the recognition system is not suitable for tasks requiring a high degree of frame accuracy. However, if our goal was such a frame-critical recognition system, COMP provides the best performance, with 70.1% $(38.1\% + 32.0\%)$, 60.5% $(23.9\% + 36.6\%)$, and 66.1% $(32.6\% + 33.5\%)$ accuracy for the user-dependent, user-independent, and user-adapted cases, respectively.
2) Looking at the charts, we see that 46% of the frames had no activity. The size of the null class is important in judging the performance of a system. In many continuous recognition tasks over 90% of the time may be the null class. Thus, the TN portion of the column

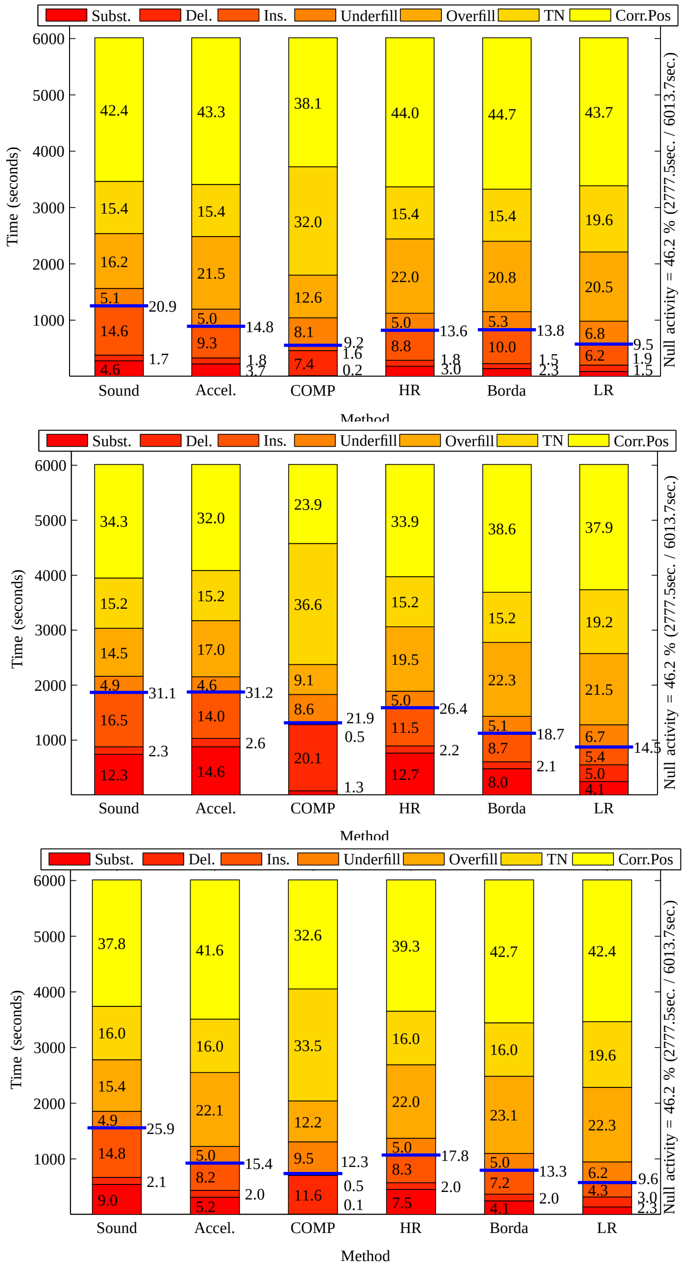


Fig. 8. Continuous results with respect to total time: correct positive, true negative (TN), Overfill, Underfill, Insertion time (Ins.), Deletion time (Del.), and Substitution time (Subst.) for the user-dependent (top), independent (middle) and adapted (bottom) cases. Serious error level is marked by the horizontal bar.

provides an implicit understanding of the type of problem being addressed. With high TN as a criteria, COMP would again be the top choice.

3) The underfill and overfill portions of the column provide an intuition of how “crisp” the recognition method is at determining activity boundaries. High levels of overfill and underfill indicate that the recognition system has difficulty determining the beginning and end of an activity, or that it breaks an activity up into smaller fragments. Thus, a researcher might once again choose COMP to minimize these errors for timing sensitive tasks.
4) The substitution, deletion, and insertion portions of the columns represent “serious errors” where the activity is completely mis-recognized. Ideally, these errors should be minimized for a recognition system intended to recognize activities as discrete events. The best performance in minimizing such errors - particularly in the user independent and adapted cases - is achieved by the logistic regression (LR) method (9.5%, 14.5% and 9.6% for the cases, respectively). In the user dependent case, COMP performs slightly better on this score (9.2%); however, unlike LR, this method does not respond well to changes in the training setup.
5) Some tasks call for a detailed analysis of the “serious errors”. If the goal is to minimize substitution and insertion errors, COMP would be the most suited according to the charts of Figure 8. If, on the other hand, it is more critical not to miss important events, keeping deletions to a minimum, one of the ranking fusion methods would be more appropriate.

V. Conclusion

We have recognized activities in a wood workshop using a heterogeneous distributed on-body sensor network consisting of microphones and accelerometers. To conclude, we discuss the relevance and limitations of our results, summarize the lessons learned, and outline future and ongoing work.

A. Limitations and Relevance

Our experiment is intended as initial exploration of continuous activity recognition using on-body sensing. In particular, we focus on activities that correspond with characteristic gestures and sounds. While our experiment involved a single, “mock” scenario, it provides insights and directions for future wearable continuous activity recognition systems. The assembly procedure involved a diverse selection of realistic activities performed by several subjects, and these activities represent a broad range of different types of sound and acceleration signatures. The combination of accelerometers and microphones for activity recognition presented in this paper seems promising for other domains. Our research groups have used similar sound recognition methods for recognizing household activities [43] and the analysis of chewing sounds [45]. We have also applied time series analysis of wrist worn accelerometers signals to American Sign Language gestures [46], bicycle repair [44], and everyday activities such as opening doors or answering the phone [47]. Given the results of these studies, we are optimistic that the techniques presented here will be valuable in these other domains.

B. Lessons Learned

a) On the use of two body worn microphones to segment continuous activities: Provided that the activities of interest are associated with a sound produced closer to the hand than to the upper arm, the strategy of using intensity differences between two separately placed microphones works relatively well for the detection of the activities. However, the strategy tends to produce short, fragmented segments. Smoothing is required to segment the data into useful events of 1-2 seconds in length. In this experiment, a successful approach classified the sound data individually in each 100ms frame using LDA and smoothed the results with a larger majority decision sliding window of 1 second. The sensitivity (recall) of this segmentation can be adjusted using a threshold on the intensity ratio difference T_{ia}. Further classification using separate sound and accelerometer based classifiers can then be performed on the discovered segments. The performance of these classifiers is affected directly by the setting of T_{ia}, and the classifiers can be tailored for specific application requirements by adjusting this parameter. For the experiments described here, this value was fixed so as to maximize the performance of positive class recognition.

b) On the combination of body-worn microphone and accelerometer features: Hand activities involving both a motion and complementary sound component can be better recognized using a fusion of classifiers (over the separate classifiers alone). For the assembly scenario investigated, the following was found:

- A simple fusion method based on comparison of outputs (COMP) provides a ‘cautious’ recognition, preferring low instances of falsely recognized activities, and almost no substitution errors (0.2% for user-dependent to 1.3% for user-independent), at the expense of more deletions than either of the constituent classifiers.
- More advanced fusion methods, based on a combination of class rankings (Borda & HR), are better at detecting all positive activities at the expense of insertions.
- The logistic regression (LR) fusion method provides a compromise in performance. This method can be trained to identify common combinations, and it produces a *NULL* result in the event of unlikely combinations. LR results in fewer insertions than Borda & HR and fewer deletions than COMP. In terms of recall and precision over positive activities, LR gives the best overall performance, ranging from 78% recall and 74% precision for the user dependent case and 66% recall and 63% precision for the user-independent case.

Note: by altering T_{ia}, the exact ratio of insertions to deletions can be adjusted according to application requirements, but in general the above holds for any fixed T_{ia} (see Figure 5).

c) On recognition robustness across different users: In user independent testing, the individual audio and gesture classifiers performed poorly compared to the user dependent scenario. However, the fused classifiers - particularly those based on class ranking - had only a relatively slight drop in performance (the COMP method became even more cautious.) Activities that allow little variation, such as the use of machine tools or tools affixed to the bench, are barely affected by changes in user. Other activities, such as the use of sandpaper or a file, allow more variation between users and consequently perform less well in user independent testing.

C. Future and Ongoing Work

We are pursuing this work in three main directions: (1) further algorithmic improvements, (2) use of different sensor combinations, and (3) application to “real-life” scenarios.

We wish to add a segmentation algorithm to the acceleration analysis and apply sensor fusion at both the classification and segmentation levels. Initial work in this direction is described in [47], [48]. We will also improve our features, particularly for acceleration, as it is clear that the information available from the arm and hand may be better combined for activity discrimination.

More detailed analysis of the sub-sequences of actions that compose the wood workshop activities should also yield improvements in performance. For example, the components of using the drill press could be modelled as “switch on,” “drill,” and “switch off.” Recognizing these sub-activities separately within the structure of an expectation grammar [49] should improve the results of recognizing the activity as a whole.

We are studying the use of ultrasonic hand tracking as a substitute for sound analysis in signal segmentation. Initial results on the utility of ultrasonic hand tracking have been described in Ogris *et al.* [44]. RFID readers to identify tools and more complex inertial sensors such as gyros and magnetometers are being investigated as additions to the sound and acceleration based system describe here.

Currently our groups are involved in a number of projects where the concepts described in this paper are used in “real-life” applications. In the WearIT@Work project, sponsored by the European Union, activity recognition is being implemented for a car assembly training task. Similar systems are planned for aircraft maintenance. In a project sponsored by the Austrian regional government of Tirol, recognition of household activities is being pursued using wrist mounted accelerometers, microphones, and other sensors. The work is ultimately envisioned as forming part of an assistive system for the cognitively disabled.

References

[1] S. Feiner, B. MacIntyre, and D. Seligmann, “Knowledge-based augmented reality,” *Com. of the ACM*, vol. 36, no. 7, pp. 52–62, 1993.

[2] M. Lampe, M. Strassner, and E. Fleisch, “A ubiquitous computing environment for aircraft maintenance,” in *ACM symp. on Applied comp.*, pp. 1586–1592, 2004.

[3] D. Abowd, A. K. Dey, R. Orr, and J. Brotherton, “Context-awareness in wearable and ubiquitous computing,” *Virtual Reality*, vol. 3, no. 3, pp. 200–211, 1998.

[4] T. Starner, B. Schiele, and A. Pentland, “Visual contextual awareness in wearable computing,” in *Proc. IEEE Int’l Symp. on Wearable Comp.*, (Pittsburgh, PA), pp. 50–57, 1998.

[5] C. Vogler and D. Metaxas, “ASL recognition based on a coupling between HMMs and 3D motion analysis,” in *ICCV*, (Bombay), 1998.

[6] A. D. Wilson and A. F. Bobick, “Learning visual behavior for gesture analysis,” in *Proc. IEEE Int’l. Symp. on Comp. Vis.*, (Coral Gables, Florida), November 1995.

[7] J. Schlenzig, E. Hunter, and R. Jain, “Recursive identification of gesture inputs using hidden Markov models,” in *2nd Conf. on Applications of Comp. Vision*, pp. 187–194, Dec. 1994.

[8] J. M. Rehg and T. Kanade, “Digiteyes:vision-based human hand tracking,” tech. rep., Carnegie Mellon, Dec 1993.

[9] J. B. J. Bussmann, W. L. J. Martens, J. H. M. Tulen, F. Schasfoort, H. J. G. van den Berg-Emons, and H. Stam, “Measuring daily behavior using ambulatory accelerometry: The activity monitor,” *Behavior Research Methods, Instrmts.+Comp.*, vol. 33, no. 3, pp. 349–356, 2001.

[10] P. Bonato, “Advances in wearable technology and applications in physical and medical rehabilitation,” *J. NeuroEngineering and Rehabilitation*, vol. 2, no. 2, 2005.

[11] K. Aminian and B. Najafi, “Capturing human motion using body-fixed sensors: outdoor measurement and clinical applications,” *Computer Animation and Virtual Worlds*, vol. 15, pp. 79–94, 2004.

[12] P. H. Veltink, H. B. J. Bussmann, W. de Vries, W. L. J. Martens, and R. C. van Lummel, “Detection of static and dynamic activities using uniaxial accelerometers,” *IEEE Trans. Rehab. Eng.*, vol. 4, no. 4, pp. 375–386, 1996.

[13] K. Aminian, P. Robert, E. E. Buchser, B. Rutschmann, D. Hayoz, and M. Depairon, “Physical activity monitoring based on accelerometry: validation and comparison with video observation,” *Med Biol Eng Comput.*, vol. 37, pp. 304–8, 1999.

[14] M. Wetzler, J. R. Borderies, O. Bigaignon, P. Guillo, and P. Gosse, “Validation of a two-axis accelerometer for monitoring patient activity during blood pressure or ecg holter monitoring,” *Clinical and Pathological Studies*, 2003.

[15] M. Uiterwaal, E. B. Glerum, H. J. Busser, and R. C. van Lummel, “Ambulatory monitoring of physical activity in working situations, a validation study,” *J Med Eng Technol.*, vol. 22, no. 4, pp. 168–72, 1998.

[16] J. Mantyjarvi, J. Himberg, and T. Seppanen, “Recognizing human motion with multiple acceleration sensors,” in *Proc. IEEE Int’l Conf. on Sys., Man. and Cybernetics*, vol. 2, pp. 747–752, 2001.

[17] C. Randell and H. Muller, “Context awareness by analysing accelerometer data,” in *Proc. IEEE Int’l Symp. on Wearable Comp.*, pp. 175–176, 2000.

[18] K. Van-Laerhoven and O. Cakmakci, “What shall we teach our pants?,” in *Proc. IEEE Int’l Symp. on Wearable Comp.*, pp. 77–83, 2000.

[19] S. Antifakos, F. Michahelles, and B. Schiele, “Proactive instructions for furniture assembly,” in *4th Int’l Conf. UbiComp*, (Gteborg, Sweden), p. 351, 2002.

[20] 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.

[21] 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*, vol. 2, pp. 1941–1944, May 2002.

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

[23] B. Clarkson, N. Sawhney, and A. Pentland, “Auditory context awareness in wearable computing,” in *Workshop on Perceptual User Interfaces*, Nov. 1998.

[24] H. Wu and M. Siegel, “Correlation of accelerometer and microphone data in the coin tap test,” *IEEE Trans. Instrumentation and Measurement.*, vol. 49, pp. 493–497, June 2000.

[25] L. Xu, A. Kryzak, and C. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” *IEEE Trans. Systems, Man., and Cybernetics*, vol. 22, pp. 418–435, May/June 1992.

[26] T. K. Ho, “Multiple classifier combination: Lessons and next steps,” in *Hybrid Methods in Pattern Recognition, World Scientific*, 2002.

[27] J. Kittler, M. Hatef, R. Duin, and J. Matas, “On combining classifiers,” *IEEE Trans. PAMI*, vol. 20, pp. 226–239, March 1998.

[28] T. K. Ho, J. J. Hull, and S. N. Srihari, “Decision combination in multiple classifier systems,” *IEEE Trans. PAMI*, vol. 16, pp. 66–75, Jan 1994.

[29] L. Bao and S. Intille, “Activity recognition from user-annotated acceleration data,” in *Pervasive, LNCS 3001*, 2004.

[30] M. Stäger, P. Lukowicz, N. Perera, T. Büren, G. Tröster, and T. Starner, “Soundbutton: Design of a low power wearable audio classification system,” in *Proc. IEEE Int’l Symp. on Wearable Comp.*, 2003.

[31] R. Duda, P. Hart, and D. Stork, *Pattern Classification, 2nd Edition.* Wiley, 2001.

[32] C. V. C. Bouten, “A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity,” *IEEE Trans. Biomed. Eng.*, vol. 44, pp. 136–147, March 1997.

[33] L. Rabiner and B. Juang, “An introduction to hidden Markov models,” *IEEE ASSP Magazine*, pp. 4–16, Jan 1986.

[34] T. Starner, J. Makhoul, R. Schwartz, and G. Chou, “On-line cursive handwriting recognition using speech recognition methods,” in *ICASSP*, pp. 125–128, 1994.

[35] K. Murphy, “The hmm toolbox for MATLAB, http://www.ai.mit.edu/ murphyk/software/hmm/hmm.html,” 1998.

[36] N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Tröster, “Wearable sensing to annotate meeting recordings,” in *Proc. IEEE Int’l Symp. on Wearable Comp.*, pp. 186–193, Oct. 2002.

[37] T. Fawcett, *ROC Graphs: Notes and Practical Considerations for Researchers.* Kluwer, 2004.

[38] E. Tapia, S. Intille, and K. Larson, “Activity recognition in the home using simple and ubiquitous sensors,” in *Pervasive, LNCS 3001*, pp. 158–175, 2004.

[39] F. Provost, T. Fawcett, and R. Kohavi, “The case against accuracy estimation for comparing induction algorithms,” in *IMLC, 15th Int’l Conf.*, 1998.

[40] I. Phillips and A. Chhabra, “Empirical performance evaluation of graphics recognition systems,” *IEEE Trans. PAMI*, vol. 21:9, pp. 849–870, 1999.

[41] A. Hoover, G. Jean-Baptiste, X. Jiang, P. Flynn, H. Bunke, D. Goldof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher, “An experimental comparison of range image segmentation algorithms,” *IEEE Trans. PAMI*, vol. 18(7), pp. 673–689, 1996.

[42] C. van Rijsbergen, *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979.

[43] 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.

[44] G. Ogris, T. Stiefmeier, H. Junker, P. Lukowicz, and G. Tröster, “Using ultrasonic hand tracking to augment motion analysis based recognition of manipulative gestures,” in *Proc. IEEE Int’l Symp. Wearable Comp.*, 2005.

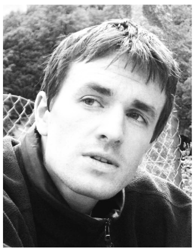
[45] O. Amft, H. Junker, and G. Tröster, “Detection of eating and drinking arm gestures using inertial body-worn sensors,” in *Proc. IEEE Int’l Symp. on Wearable Comp.*, Oct. 2005.

[46] H. Brashear, T. Starner, P. Lukowicz, and H. Junker, “Using multiple sensors for mobile sign language recognition,” in *Proc. IEEE Int’l Symp. on Wearable Comp.*, (White Plains, NY), pp. 45–53, 2003.

[47] H. Junker, P. Lukowicz, and G. Tröster, “Continuous recognition of arm activities with body-worn inertial sensors.,” in *Proc. IEEE Int’l Symp. on Wearable Comp.*, pp. 188–189, 2004.

[48] 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.

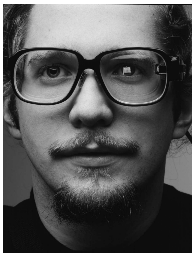
[49] D. Minnen, I. Essa, and T. Starner, “Expectation grammars: Leveraging high-level expectations for activity recognition,” in *IEEE Proc. Comp. Vision and Pattern Rec.*, June 2003.



Jamie A Ward (S'01) received the B.Eng. degree with joint honours in computer science and electronics from the University of Edinburgh, Scotland, U.K., in 2000. He spent his first year after graduation working as an Analogue Designer for a start-up electronics firm in Austria before joining the Swiss Federal Institute of Technology (ETH), Wearable Computing Laboratory in Zürich. He is currently working towards the Ph.D. degree at ETH, which is due for completion in Spring 2006. His current research is divided between work on activity recognition using heterogeneous on-body sensors, and methods for evaluating performance in continuous recognition systems.

Paul Lukowicz (M'96) received the M.Sc. degree in computer science, the M.Sc. degree in physics, and the Ph.D. degree in computer science from the University of Karlsruhe, Germany, in 1992, 1993, and 1999, respectively. From 1999-2004, he was in charge of the Wearable Computing Laboratory and the Computer Architecture Group at the Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology (ETH) Zürich. Between 2003 and 2006, he was a Professor of Computer Science at the University of Health Informatics and Technology Tirol, Innsbruck, Austria. Since April 2006 he is a Full Professor at the University of Passau, Germany where he has the Chair for Embedded Systems and Pervasive Computing. His research interests include wearable and mobile computer architecture, context and activity recognition, high-performance computing, and optoelectronic interconnection technology.

Gerhard Tröster (SM'93) received the M.Sc. degree from the Technical University of Karlsruhe, Germany, in 1978, and the Ph.D degree from the Technical University of Darmstadt, Germany, in 1984, both in electrical engineering. Since 1993 he is a Professor and head of the Electronics Laboratory, Swiss Federal Institute of Technology (ETH) Zürich. During the eight years he spent at Telefunken Corporation (atmel), Germany, he was responsible for various national and international research projects focused on key components for ISDN and digital mobile phones. His field of research includes wearable computing, smart textiles, electronic packaging and miniaturized digital signal processing. He authored and coauthored more than 100 articles and holds five patents. In 1997, he co-founded the spin-off u-blox AG.



Thad E Starner Thad Starner is an Assistant Professor at the Georgia Institute of Technology College of Computing. He holds four degrees from MIT, including his PhD from the MIT Media Laboratory in 1999. He is a wearable computing pioneer, having worn a wearable as an everyday personal assistant since 1993. Thad publishes on mobile human computer interfaces, intelligent agents, computer vision, and augmented reality, and his work focuses on computational assistants for everyday-use wearable computers.