Gesture Spotting Using Wrist Worn Microphone and 3-Axis Accelerometer

Jamie A. Ward[1], *Paul Lukowicz*[2] *and Gerhard Tröster*[1]

[1] Swiss Federal Institute of Technology (ETH), Wearable Computing Lab, Zurich
[2] UMIT (University of Health Sciences, Medical Informatics and Technology, Hall i. Tirol, Austria

Abstract

We perform continuous activity recognition using only two wrist-worn sensors - a 3-axis accelerometer and a microphone. We build on the intuitive notion that two very different sensors are unlikely to agree in classification of a false activity. By comparing imperfect, jumping window classifications from each of these sensors, we are able discern activities of interest from null or uninteresting activities. Where one sensor alone is unable to perform such partitioning, using comparison we are able to report good overall system performance of up to 70% accuracy. In presenting these results, we attempt to give a more-in depth visualization of the errors than can be gathered from confusion matrices alone.

1. Introduction

Hand actions play a crucial role in most human activities. As a consequence, detecting and recognising such activities is an important aspect of context recognition. It is also one of the most difficult. This is particularly true for continuous recognition where a set of relevant hand motions (gestures) need to be spotted in a data stream. The difficulties of such recognition stem from two things. First, due to a large number of degrees of freedom, hand motions tend to be very diverse. The same activity might be performed in many different ways even by a single person. Second, in terms of motion, hands are the most active body parts. We move our hands continuously, mostly in an unstructured way, even when not doing anything particular with them. In fact in most situations such unstructured motions by far outnumber gestures that are relevant for context recognition. This means that a continuous gesture spotting applications has to deal with a zero, or *NULL*, class that is difficult to model while taking up most of the signal.

1.1 Paper Contributions

Our group has invested a considerable amount of work into hand gesture spotting. To date this work has focused on using several sensors distributed over the user's body with the aim of maximising recognition performance. This included motion sensors (3 axis accelerometer, 3 axis gyroscopes and 3 axis magnetic sensors) on the upper and lower arm [5], microphone/accelerometer combination on the upper and lower arm [7] as well as, more recently, a combination of several motion sensors and ultrasonic location devices [10].

This paper investigates the performance of a gesture spotting system based on a single, wrist mounted device. The idea behind the work is that wrist mounted accessories are broadly accepted and worn by most people on daily basis. In contrast, systems that require the user to put on several sensors at locations such as the upper arm may have problems with user acceptance.

The downside of this approach is the reduced amount of information available for the recognition. This means, for example, that the method of analysing sound intensity differences between microphones on different parts of the body, which was the cornerstone of our previous signal partitioning work, is not feasible. It is also important that the devices used have small form factor and thus do not require too much computing power so as to keep battery size small.

The main contribution of the paper is to show that, for a certain subset of hand based activities - the use of tools in a woodwork assembly scenario -, reasonable gesture spotting results can be achieved using only a combination of microphone and 3 axis accelerometer mounted on the wrist.

The method relies on simple jumping window sound processing algorithms that have been shown [15] to require only minimal computational and communication performance.

For the acceleration, inference on Hidden Markov Models (HMM) is used, again on jumping windows across the data. The results from the two classifiers (sound and acceleration) are then combined to produce a final output. The aim is to show that although individual sensor classifiers have no way of separating valid activities from *NULL*, their combination provides a means of doing so.

This approach is verified using data which was gathered from an extended, multi-subject, run of the wood workshop assembly experiment first introduced in [7]. The results are presented using both traditional confusion matrices, plus a novel visualisation method that provides a more in-depth understanding of the error types.

1.2 Related Work

Most of the existing work on gesture recognition involves the use of computer vision [17, 19, 20, 14]. Regarding non-visual sensors, previous setups and algorithms have proved successfull either for segmented recognition, or for scenarios where the *NULL* class was easy to model or not relevant (e.g. recognition of standing, sitting, walking, running [8, 13, 18] using acceleration sensors.) In the work of [11, 2] sound was exploited for performing situation analysis. Sound was also used in [1] to improve the performance of hearing aids. Complimentary information from sound and acceleration has been used before to detect defects in material surfaces [21], but no work, of which the authors are aware, use these for recognition of complex activities.

2. Experiment

The dataset was gathered from experiments based on a mock assembly scenario involving the use of hand held and hand operated tools and machines in a wood workshop. This setup was previously used for our earlier study on activity recognition, and was chosen as a suitable testbench for the continuing work into the development of wearable computers for maintenance and assembly applications [1]. Though obtained from a fairly constrained environment, the diverse selection of hand and tool activities provides a useful dataset for evaluating activity recognition techniques. Gestures involving hand interaction with tools

[1] The development of such systems is the aim of the European Union WearIT@Work project in which our group participates.

Figure 1: The wood workshop with *(1)* grinder, *(2)* drill, *(3)* file and saw, *(4)* vise, and *(5)* cabinet with drawers

generally have both a characteristic motion and a corresponding sound component, from which recognition using these two different sensing modalities is particularly suited.

Figure 1 shows the environment and tools used. The 9 activities which we set out to spot were: hammering, sawing, filing, using a machine drill, sanding, using a machine grinder, screwdriving, opening and closing a vise, and opening and closing a drawer. All other activities and movements were labelled as *NULL*.

Specifically, the assembly sequence consisted of sawing a piece of wood, drilling a hole in it, grinding a piece of metal, attaching it to the piece of wood with a screw, hammering in a nail to connect the two pieces of wood, and then finishing the product by smoothing away rough edges with a file and a piece of sandpaper. The wood was fixed in the vise for sawing, filing, and smoothing (and removed whenever necessary). The test subject moved between areas in the workshop between steps. Also, whenever a tool or an object (nail screw, wood) was required, it was retrieved from its drawer in the cabinet and returned after use.

The first dataset, as reported in the earlier study, involved only a single subject performing this sequence. The experiment has since been revised to include more subjects (1 female and 4 male), with each subject repeating the sequence between 3 and 6 times, thus producing a total dataset of (3+3+4+4+6)=20 recordings. (Some subjects performed more repetitions than others due to a combination of technical problems in recording and availability.) Each sequence lasted on average five minutes, bringing the total dataset size to 6014 seconds.

The data was collected using a Sony microphone and a 3-axis accelerometer (from the ETH PadNET sensor network [6]). These were strapped to each subject's right wrist - in the current set, all subjects are right handed. (Readings were also taken from each subject's upper arm, though this data is not used here.)

3. Recognition Method

We apply jumping windows of length w_{len} seconds across all the data in increments of w_{jmp}. At each step we apply an LDA based classification on the sound data, and an HMM classification on the sound. The 'soft' results of each classification - LDA distances for sound and HMM class likelihoods for acceleration - are converted into class rankings, and these are fused together using one of two methods: comparison of top rank (COMP), and a method using Logistic Regression (LR).

3.1 Frame by Frame Sound Classification Using LDA

Frame-by-frame sound classification was carried out using pattern matching of features extracted in the frequency domain. Each frame represents a window on 100ms of raw audio data. These windows are then jumped over the entire dataset in 25ms increments, producing a 40Hz output.

The audio stream was taken at a sample rate of 2kHz from the wrist worn microphone. From this a Fast Fourier Transform (FFT) was carried out on each 100ms window, generating a 100 bin output vector $(1/2 * fs * fftwnd = 1/2 * 2 * 100 = 100bins)$.

Making use of the fact that our recognition problem requires a small finite number of classes, we applied Linear Discriminant Analysis (LDA)[3] to reduce the dimensionality of these FFT vectors from 100 to $\#Classes - 1$.

Classification of each frame can then be carried out using a simple Euclidean minimum distance calculation. Whenever we wish to make a decision, we simply calculate the incoming point in LDA space and find its nearest class mean value from the training dataset. This saving in computation complexity by dimensionality reduction comes at the comparatively minor cost of requiring us to compute and store a set of LDA class mean values from which the LDA distances might be obtained.

Equally, a nearest neighbour approach might be used. For the experiment described here however, Euclidean distance was found to be sufficient.

A larger window, w_{len}, was moved over the data in w_{jmp} second increments. This relatively large window was chosen to reflect the fact that all of the activities we are interested in occur at the timescale of at least several seconds. On each window we compute a sum of the constituent LDA distances for each class. From these total distances, we then rank each class according to minimum distance. Classification of the window is then simply a matter of choosing the top ranking class.

3.2 HMM Acceleration Classification

In contrast to the approach used for sound recognition, we employed model based classification, specifically the Hidden Markov Model (HMM), for classifying accelerometer data[12, 16]. (The implementation of the HMM learning and inference routines for this experiment was provided courtesy of Kevin P. Murphy's HMM Toolbox for matlab [9].)

The features used to feed the HMM models were calculated from jumping 100ms windows on the x,y, and z axis of the 100Hz sampled acceleration data. These windows were moved over the data in 25ms increments, producing the following features, output at 40Hz:

- Mean of x-axis
- Variance of x-axis
- A count of the number of peaks (for x,y,z)
- Mean amplitude of the peaks (for x,y,z)

Finally we globally standardised the features so as to avoid numerical complications with the model learning algorithms in matlab.

In previous work we employed single Gaussian observation models, but this was found to be inadequate for some classes unless a large number of states were used. Intuitively, the descriptive power of a mixture of Gaussian is much closer to 're-ality' than only one, and so for these classes a mixture model was used. The specific number of mixtures and the number of hidden states used were individually tailored by hand for each class. The parameters were obtained from the data using leave-one-out training.

A window of w_{len}, in w_{jmp} increments, was run over the acceleration features, and the corresponding log likelihood for each HMM class model calculated.

Classification is carried out for each window by choosing the class which produces the largest log likelihood given the stream of feature data from the test set.

3.3 Fusion of classifiers

3.3.1 Comparison of top choices (COMP)

The top rankings from each of the sound and acceleration classifiers for a given jumping window segment are taken, compared, and returned as valid if they agree. Those where both classifiers disagree are thrown out - classified as null.

3.3.2 Logistic regression (LR)

The main problem with a direct comparison of top classifier rankings is that it fails to take into account cases where one classifier might be more reliable than another at recognising particular classes. If one classifier reliably detects a class, but the other classifier fails to, perhaps relegating the class to second or third rank, then a basic comparison would just assign null. For such cases, then a 'softer' method of classifier fusion is needed - one that takes into account the different rankings of each classifier.

In the work of Ho et. al. [4], three methods for classifier fusion based on class rankings are presented and evaluated: Highest Rank, whereby each class is assigned a rank according to the highest rank assigned to it by any of the classifiers; Borda count, whereby each class is ranked according to the total number of classes ranking below it by each classifier; and Logistic Regression (LR), a method based on the Borda count, but which estimates weights for each class combination using regression.

Of the methods presented, only one of them, the Logistic Regression (LR) makes sense to apply here, as it is the only one which provides the scope to deal with assigning results to null.

The basic motivation behind LR is to assign a score for each class and every combination of classifier rankings. However, such a scoring would soon become computationally prohibitive, even for a moderate number of classes and classifiers. Instead, LR makes use of 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. So for each class a function is 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, β the logistic regression coefficients. These coefficients are computed by applying a suitable regression fit using the correctly classified ranking combinations from the training data. Again the training is performed on a leave-one-out basis.

So that unlikely combinations are assigned to null, we introduce an empirically obtained threshold on $L(x)$ for each class. Of the classes which fall below this threshold, the most likely $L(x)$ value is taken and re-assigned to the 'null class'. This means that if all classes fall below their threshold for a given ranking combination, then the null will take top ranking.

Classification can then be carried out by estimating L(X) for each class on the input rankings, comparing with the null threshold, and then ranking the values obtained. The final classification result can then be taken from the highest rank.

4. Results

The system was initially evaluated across sweeps of the two main parameters, window length w_{len} and window jump length w_{jmp}. From these sweeps, setting both w_{len} and w_{jmp} to 2 seconds was found to produce favourable results. All further analysis was carried out with these parameters set.

Both the LDA and HMM methods 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.

We applied HMM classification to the accelerometer data, and LDA minimum distance to the audio. This was applied to all 20 sets of data. Typical results from one of these sets is plotted in Figures 2, with class predictions compared alongside the hand-labelled ground truth.

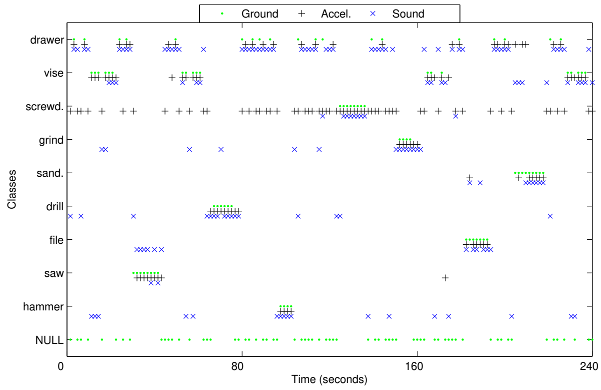

Figure 2: Plot of a typical classifier output sequence showing the ground truth, the sound predictions, and acceleration predictions (1 output sample every 2 seconds)

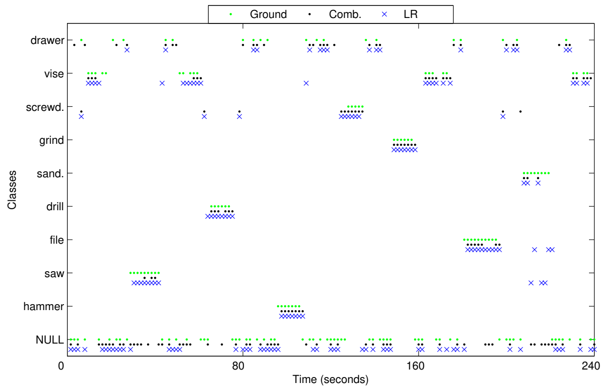

Figure 3: Plot of output sequences of sound and acceleration predictions combined using Comparison method, and LR, versus ground truth

With each of the 2 second segments, we performed firstly the classification comparison fusion, and then the logistic regression using the rankings obtained from the HMM likelihood and LDA distance information.

On first run, the LR method continued to produce a large number of insertions - primarily from the class 'screwdriving'. This was due to the fact that this is comparatively silent class, and as the training data consisted mostly of noisy, positive class examples (at no stage do we use *NULL* labelled data for training), it winds up being a 'catch all' class for non-activities which should have been assigned *NULL*. Reducing the weights of the ranking combinations for this class during training helps to alleviate this problem.

The final predictions from each of these, compared alongside the ground truth, are shown in 3.

Lacking any ability to distinguish valid activities from *NULL*, the constituent classifiers, as expected, produce much noise. With LDA tending to misclassify *NULL* as a quiet class, such as screwdriving; and HMM generally giving random misclassifications. Both perform relatively well when set against known system classes however, and this is reflected in the performance of both the comparison and LR predictions.

Plotting predictions might allow us to gain a rough understanding of how well the system performs for a given set, but for a measure across all the data we require a more quantitative means. For this we perform a direct frame by frame comparison of the predictions with the ground truth, and fill out a confusion matrix of the results. We sum the matrices across all test datasets and present the total matrix, for each recognition method, in Tables 1. Class by class recognition rates, stating how well the system returns true frames are given to the right of these tables. Also shown is a summary of the False Positive (FP), False negative (FN), Substitution, Correct True Positive (cTP) and the overall Accuracy as percentages of the total experiment time. (Substitution being defined as the misclassification of one positive, non-*NULL*, class for another; and cTP as the correctly classified positive class.) This summary information is also shown, in barchart form, in Figure 4.

Continuous recognition systems which deal with human activity are often characterised by the lack of fixed, well-defined activity boundaries. In many cases, whether an activity was recognised exactly within the labelled time frame, or slightly off from it, is less important than the fact that the activity was detected correctly in the first place. The confusion matrix based evaluation as given does not account for such 'fuzzy' boundaries, and makes a strict judgement on the predicted frames according to the given ground truth.

If we lighten this restriction, we can create two additional error classifications, which we call $overfill$ and $underfill$, as defined:

- Overfill time: when a continuous sequence of correct prediction frames slips over the ground truth boundary to cover *NULL* labelled frames (previously classed as insertion time)
- Underfill time: the time left when a continuous sequence of correct prediction frames does not completely cover the corresponding ground truth (previously classed as deletion time)

Taking account of this, the total overfill and underfill, together with substitution, deletion, insertion, correct positive and correct negatives times as a percentage of the overall experiment, are shown in Figure 5. To mark the level of true insertion, deletion and substitution errors, we introduce a 'serious error' measure, as shown on the charts.

5. Discusion

As expected, the individual recognition performance for each of the two sensor types performed quite poorly on their own, but once combined the results improved dramatically.

As a percentage of the entire time, substitution errors decreased from a maximum of 9.3% by HMM on acceleration to as low as 0.6% in the basic Comparison method (and a respectable 2.3% for LR).

The amount of false positives as a percentage of total time fell to 14.9% for the Comparison method. LR, which although having more (25.7%) false positives, is, however, the better choice for fewer false negatives (6% LR, versus 14.4% for Comparison).

When underfill and overfill are considered, these results begin to take on new meaning, as the more serious errors of insertions and deletions prove to occur far less than the count of false positives and false negatives might suggest. As a percentage of the total time, the sum of insertion, deletion and substitution errors is only around 7% for the Comparison and 9% for LR methods.

As with any comparison between recognition systems, it is unwise to make claims as to the absolute superiority of one method over the other - the differences between basic Comparison and LR should be highlighted in view of whatever performance criteria is most important to the application. Some applications, safety monitoring of dangerous activities for example, might regard a false negative error as being much worse than a false positive. In which case the LR method as given might be regarded preferable.

Additionally, the parameters which have for the purposes of this paper been set to some 'optimal' value, such as the *NULL* thresholds on L(x) for LR, can alter the nature of these results by raising or lowering the chance of returning a *NULL*. It is, for example, possible to tailor the LR method to have exactly the same performance as Comparison if one raises the threshold to just under the L(x) value for matching top rank classifier results. This ability means that although more complex to implement, the LR is more versatile in terms of performance optimisation than the basic comparison.

The purpose of this paper, however, is not to analyse the peculiarities of each method in depth (one might use ROC curves for this purpose), but rather to evaluate the feasibility of their useage in discerning useful activities from *NULL* in a recognition task where two different sensor modalities, neither of which can perform this task alone, are used.

5.1 Conclusion

Using only a single wrist worn unit containing two sensors - a microphone and a 3-axis accelerometer - it is possible to perform gesture spotting for a certain subset of activities. Recognition of activities is carried out for each sensor using standard jumping window based approaches. Alone, neither sensor can detect a *NULL* gesture, but when fused together, this becomes possible.

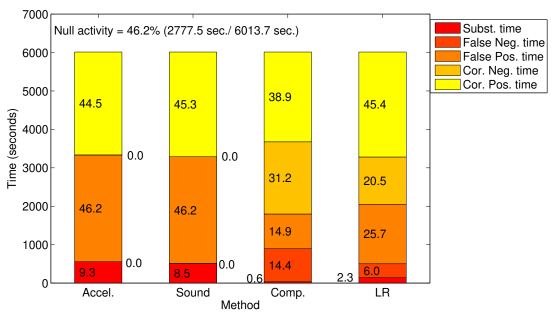

Figure 4: Breakdown of errors as a percentage of total experiment time for acceleration, sound and combined: Correct Positive, Correct Negative, False Positive, False Negative and Substitution times, as taken directly from the confusion matrix

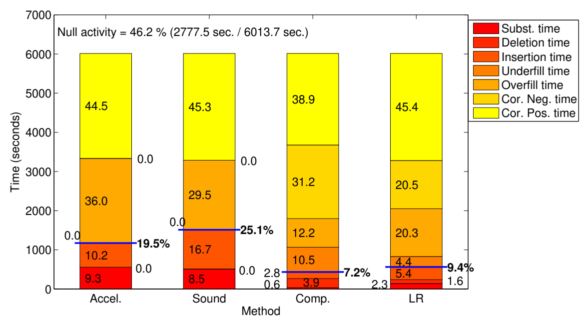

Figure 5: Breakdown of errors as a percentage of total experiment time for acceleration, sound and combined: Correct Positive, Correct Negative, Overfill, Underfill, Insertion, Deletion and Substitution times; also given is the 'serious error' level, which ignores the minor errors of Overfill and Underfill

Class	Total(s)	hammer	saw	file	drill	sand	grind	screwdr.	vise	drawer	*NULL*	%Correct
hammer	195.7	**183.6**			2.5		3.6	0.6	2.4	3.0		93.84
saw	306.4	3.7	**209.9**	69.7	3.9	4.0			15.2	0.1		68.50
file	304.6	0.6	40.3	**248.3**	1.9	8.0		0.3	4.9	0.2		81.52
drill	241.5	0.9			**184.0**		50.6		2.0	4.0		76.20
sand	313.0	0.7	3.3	40.5	6.3	**228.2**		2.7	29.6	1.8		72.91
grind	277.7				15.1		**260.6**		2.0			93.85
screwdr.	260.4	19.3			2.0		2.0	**229.7**		7.3		88.23
vise	678.1	65.2			28.5	0.4	11.4	4.6	**543.4**	24.6		80.14
drawer	658.8	8.2			30.3		10.9	7.4	12.0	**590.1**		89.57
NULL	2777.5	185.9	12.5	13.5	471.5	1.8	304.9	77.2	296.3	1413.9		0

Accel.	Total: 6013.7	FN: 0.0	FP: 2777.5	Subst.: 558.2	cTP: 2678.0	cTP+TN: 2678.0
		0.0%	46.2%	9.3%	44.5%	Accuracy: **44.5%**

Class	Total(s)	hammer	saw	file	drill	sand	grind	screwdr.	vise	drawer	*NULL*	%Correct
hammer	195.7	**168.5**						19.6	6.7	0.8		86.12
saw	306.4		**267.2**	14.0		2.0		10.5	12.7			87.21
file	304.6			**238.8**		34.4	1	16.1	2.5	2.8		78.40
drill	241.5				**226.5**	12.0	2.0	1.0				93.79
sand.	313.0		6.0	13.9		**258.0**	2.0	21.7	0.9	10.5		82.42
grind.	277.7					2.9	**274.5**	0.3				98.86
screwdr.	260.4							**249.0**	9.8	1.6		95.64
vise	678.1		0.3					101.8	**571.2**	4.8		84.24
drawer	658.8					0.7		163.6	22.7	**471.8**		71.61
NULL	2777.5	5.5	8.8	7.3	111.5	18.0	67.5	1360.8	506.4	691.7		0

Sound	Total: 6013.7	FN: 0.0	FP: 2777.5	Subst.: 510.6	cTP: 2725.6	cTP+TN: 2725.6
		0.0%	46.2%	8.5%	45.3%	Accuracy: **45.3%**

Class	Total(s)	hammer	saw	file	drill	sand	grind	screwdr.	vise	drawer	*NULL*	%Correct
hammer	195.7	**168.5**						0.6	1.3	0.8	24.5	86.12
saw	306.4		**200.3**	12.0					5.0		89.1	65.38
file	304.6			**203.3**		2.0		0.3			99.1	66.73
drill	241.5				**169.0**						72.5	69.99
sand.	313.0					**194.6**		1.1	0.9		116.4	62.16
grind.	277.7						**259.5**				18.2	93.45
screwdr.	260.4							**225.7**			34.6	86.70
vise	678.1							1.7	**476.2**	1.0	199.2	70.23
drawer	658.8							5.4	2.1	**440**	211.3	66.79
NULL	2777.5	3.5	1.7	2.7	83.0	1.4	50.5	67.2	126.1	562.2	**1879.2**	67.66

Comp.	Total: 6013.7	FN: 864.9	FP: 898.3	Subst.: 34.1	cTP: 2337.1	cTP+TN: 4216.4
		14.4%	14.9%	0.6%	38.9%	Accuracy: **70.1%**

Class	Total(s)	hammer	saw	file	drill	sand	grind	screwdr.	vise	drawer	*NULL*	%Correct
hammer	195.7	**169.5**						0.6	4.1	2.2	19.4	86.61
saw	306.4		**254.3**	22.0					9.4	0.1	20.6	83.00
file	304.6		4.0	**262.2**		12.0		0.3	3.1	0.2	22.9	86.07
drill	241.5				**232.3**		4.2				5.0	96.19
sand.	313.0		8.0	22.8		**237.6**		1.1	8.8	1.1	33.7	75.89
grind.	277.7						**275.6**				2.1	99.26
screwdr.	260.4							**225.7**		2.8	31.8	86.70
vise	678.1		0.3					1.7	**566.7**	17.7	91.7	83.58
drawer	658.8							5.4	8.1	**508.5**	136.8	77.19
NULL	2777.5	10.5	9.7	7.1	121.7	2.4	76.2	67.2	282.8	969.4	**1230.6**	44.30

LR	Total: 6013.7	FN: 363.8	FP: 1546.9	Subst.: 139.9	cTP: 2732.5	cTP+TN: 3963.1
		6.0%	25.7%	2.3%	45.4%	Accuracy: **65.9%**

Table 1: Confusion matrices for the acceleration and sound classifications, and the comparison (Comp.) and logistic regression (LR) combinations, with jumping window of 2 seconds. The total % Correct is a summation of the class correct times over the total time. All times are given in seconds. At the bottom of each matrix, a summary table gives times and percentages of false negative (FN), false positive (FP), substitution (Subst.), correct true positive (cTP), and overall correct (cTP+cTN), corresponding to the information in Figure 4.

It has been shown that this setup is particularly suited to recognising assembly-type activities, involving use of hand manipulated machines and tools. Clearly, sound-acceleration combination might not be useful for gestures which produce little or no sound, such as in sign language. However, in applications involving the use of hand-held objects which produce both motion and corresponding sound components these methods are feasible.

In evaluating recognition performance, we introduce the terms 'underfill' and 'overfill' to describe those common cases in continuous recognition where events fail to completely match the ground truth - but which might actually be judged correct by a human observer - and show how these can be used to visualise results. By discounting overfill and underfill errors, the lowest error rates for the described system fall from around 30% to 7.2%.

A remaining issue, which is left here for future work, is the influence of background noise on the sound recognition. It is the belief of the authors that although this might be a limiting factor, the careful selection and placement of microphones should help mitigate the effects - especially for recognition of dominant, loud activities such as hammering, or sawing.

6. Acknowledgements

The authors would like to thank Thad Starner from Georgia Institute of Technology for his invaluable advice and support in this work.

7. References

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

[2] B. Clarkson, N. Sawhney, and A. Pentland. Auditory context awareness in wearable computing. In *Workshop on Perceptual User Interfaces*, November 1998.

[3] R. Duda, P. Hart, and D. Stork. *Pattern Classification, Second Edition.* Wiley, 2001.

[4] Tin Kam Ho, J.J. Hull, and S.N Srihari. Decision combination in multiple classifier systems. In *IEEE TPAMI*, volume 16, pages 66–75, Jan 1994.

[5] Holger Junker, Paul Lukowicz, and Gerhard Tröster. Continuous recognition of arm activities with body-worn inertial sensors. In *ISWC*, pages 188–189, 2004.

[6] N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Tröster. Wearable sensing to annotate meeting record-

ings. In *IEEE Int'l Symp. on Wearable Comp.*, pages 186–193, October 2002.

[7] Paul Lukowicz, Jamie A Ward, Holger Junker, Gerhard Tröster, Amin Atrash, and Thad Starner. Recognizing workshop activity using body worn microphones and accelerometers. In *Pervasive, LNCS 3001*, 2004.

[8] J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with multiple acceleration sensors. In *2001 IEEE Int'l Conf. on Systems, Man and Cybernetics*, volume 3494, pages 747–752, 2001.

[9] Kevin P. Murphy. The hmm toolbox for MATLAB, http://www.ai.mit.edu/ murphyk/software/hmm/hmm.html.

[10] Georg Ogris, Thomas Stiefmeier, Holger Junker, Paul Lukowicz, and Gerhard Trster. Using ultrasonic hand tracking to augment motion analysis based recognition of manipulative gestures. In *(to appear in) Proc. 9th IEEE Int. Symp. on Wearable Computers*, Osaka, Japan, 2005. IEEE.

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

[12] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–16, January 1986.

[13] C. Randell and H. Muller. Context awareness by analysing accelerometer data. In *IEEE Int'l Symp. on Wearable Comp.*, pages 175–176, 2000.

[14] J. M. Rehg and T. Kanade. Digiteyes: vision-based human hand tracking. Technical report, Carnegie Mellon University, Dec 1993.

[15] Mathias Stäger, Paul Lukowicz, Gerhard Tröster, and Thad Starner. Implementation and evaluation of a low-power sound-based user activity recognition system. 8th Int'l Symp. on Wearable Comp., 2004.

[16] T. Starner, J. Makhoul, R. Schwartz, and G. Chou. On-line cursive handwriting recognition using speech recognition methods. In *ICASSP*, pages 125–128, 1994.

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

[18] K. Van-Laerhoven and O. Cakmakci. What shall we teach our pants? In *IEEE Int'l Symp. on Wearable Comp.*, pages 77–83, 2000.

[19] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *ICCV*, Bombay, 1998.

[20] 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.

[21] Huadong Wu and Mel Siegel. Correlation of accelerometer and microphone data in the coin tap test. In *Instrumentation and Measurement, IEEE Trans.*, volume 49, pages 493–497, June 2000.