Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers

Paul Lukowicz[1], Jamie A. Ward[1], Holger Junker[1], Mathias Stäger[1], Gerhard Tröster[1], Amin Atrash[2], and Thad Starner[2]

[1] Wearable Computing Laboratory, ETH Zürich
8092 Zürich, Switzerland, www.wearable.ethz.ch
{lukowicz,ward}@ife.ee.ethz.ch
[2] College of Computing, Georgia Institute of Technology
Atlanta, Georgia 30332-0280
{amin,thad}@cc.gatech.edu

Abstract. The paper presents a technique to automatically track the progress of maintenance or assembly tasks using body worn sensors. The technique is based on a novel way of combining data from accelerometers with simple frequency matching sound classification. This includes the intensity analysis of signals from microphones at different body locations to correlate environmental sounds with user activity.
To evaluate our method we apply it to activities in a wood shop. On a simulated assembly task our system can successfully segment and identify most shop activities in a continuous data stream with zero false positives and 84.4% accuracy.

1 Introduction

Maintenance and assembly are among the most important applications of wearable computing to date; the use of such technology in tasks such as aircraft assembly [17], vehicle maintenance [4] and other on-site tasks [2,7] demonstrates a genuine utility of wearable systems.

The key characteristic of such applications is the need for the user to physically and perceptually focus on a complex real world task. Thus in general the user cannot devote much attention to interaction with the system. Further the use of the system should not restrict the operators physical freedom of action. As a consequence most conventional mobile computing paradigms are unsuitable for this application field. Instead wearable systems emphasizing physically unobtrusive form factor, hands free input, head mounted display output and low cognitive load interaction need to be used.

Our work aims to further reduce the cognitive load on the user while at the same time extending the range of services provided by the system. To this end we show how wearable systems can automatically follow the progress of a given maintenance or assembly task using a set of simple body worn sensors. With such *context* knowledge the wearable could pro-actively provide assistance without the need for any explicit action by the user. For example, a maintenance

A. Ferscha and F. Mattern (Eds.): PERVASIVE 2004, LNCS 3001, pp. 18–32, 2004.

support system could recognize which particular subtask is being performed and automatically display the relevant manual pages on the system's head-up display. The wearable could also record the sequence of operations that are being performed for later analysis, or could be used to warn the user if an important step has been missed.

1.1 Related Work

Many wearable systems explore *context* and proactiveness (e.g [1]) as means of reducing the cognitive load on the user. Much work has also been devoted to recognition methods, in particular the use of computer vision [20,24,25,16,15].

The application of proactive systems for assisting basic assembly tasks has been explored in [22], however this is built on the assumption of sensors integrated into the objects being assembled, not on the user doing the assembly.

Activity recognition based on body worn sensors, in particular acceleration sensors, has been studied by different research groups [11,14,23]. However all of the above work focused on recognizing comparatively simple activities (walking, running, and sitting). Sound based situation analysis has been investigated by Pelton *et al.* and in the wearables domain by Clarkson and Pentland [12, 5]. Intelligent hearing aids have also exploited sound analysis to improve their performance [3].

1.2 Paper Aims and Contributions

This paper is part of our work aiming to develop a reliable context recognition methodology based on simple sensors integrated in the user's outfit and in the user's artifacts (e.g. tools, appliances, or parts of the machinery) [10]). It presents a novel way of combining motion (acceleration) sensor based gesture recognition [8] with sound data from distributed microphones [18]. In particular we exploit intensity differences between a microphone on the wrist of the dominant hand and on the chest to identify relevant actions performed by the user's hand.

In the paper we focus on using the above method to track the progress of an assembly task. As described above such tasks can significantly benefit from activity recognition. At the same time they tend to be well structured and limited to a reasonable number of often repetitive actions. In addition, machines and tools typical to a workshop environment generate distinct sounds. Therefore these activities are well suited for a combination of gesture and sound–based recognition.

This paper describes our approach and the results produced in an experiment performed on an assembly task in a wood workshop. We demonstrate that simple sensors placed on the user's body can reliably select and recognize user actions during a workshop procedure.

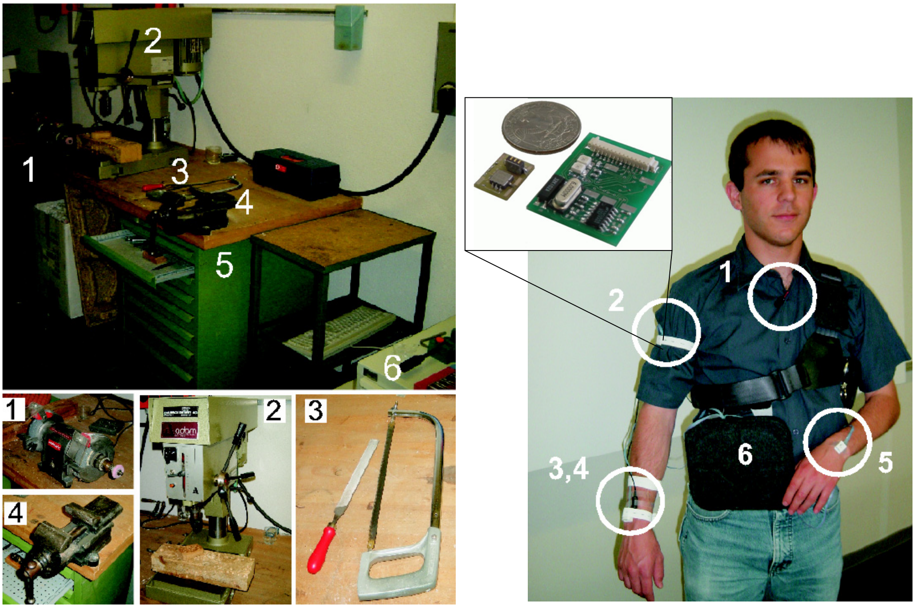

Fig. 1. The wood workshop (*left*) with *(1)* grinder, *(2)* drill, *(3)* file and saw, *(4)* vise, and *(5)* cabinet with drawers. The sensor type and placement (*right*): *(1,4)* microphone, *(2,3,5)* 3-axis acceleration sensors and *(6)* computer

2 Experimental Setup

Performing initial experiments on live assembly or maintenance tasks is inadvisable due to the cost and safety concerns and the ability to obtain repeatable measurements under experimental conditions. As a consequence we have decided to focus on an "artificial" task performed at the workbench of wood workshop of our lab (see Figure 1). The task consisted of assembling a simple object made of two pieces of wood and a piece of metal. The task required 8 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.

2.1 Procedure

The assembly sequence consists 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 be-

Table 1. Steps of workshop assembly task

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 the hammer, get the driver and the 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 the 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

tween 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 exact sequence of actions is listed in Table 1. The task was to recognize all tool-based activities. Tool-based activities exclude drawer manipulation, user locomotion, and clapping (a calibration gesture). The experiment was repeated 10 times in the same sequence to collect data for training and testing. 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.

2.2 Data Collection System

The data was collected using the ETH PadNET sensor network [8] equipped with 3 axis accelerometer nodes and two Sony mono microphones connected to a body worn computer. The position of the sensors on the body is shown in Figure 1: an accelerometer node on both wrist and on the upper arm of the right hand, and a microphone on the chest and on the right wrist (the test subject was right handed).

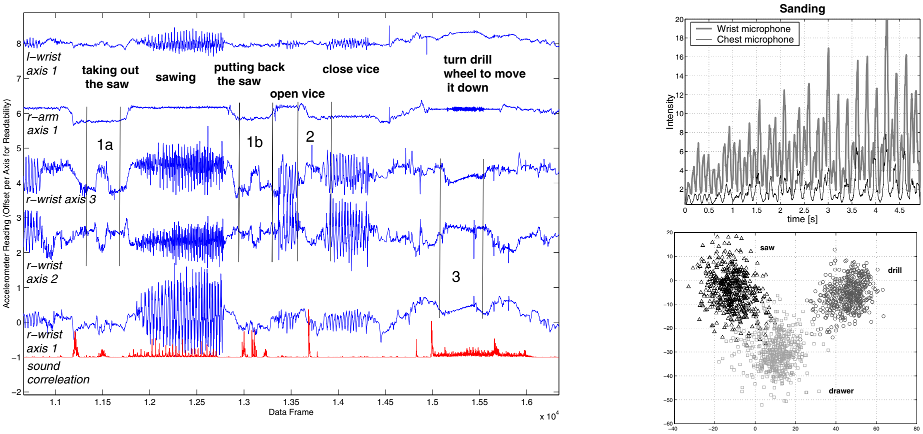

Fig. 2. Example accelerometer data from sawing and drilling (*left*); audio profile of sanding from wrist and chest microphones (*top right*); and clustering of activities in LDA space (*bottom right*)

As can be seen in Figure 1 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 6 MHz maximum clock speed. The current module version reads out up to 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 ($f_{cutoff} = 50Hz$) and digitized with 12Bit resolution using a sampling rate of 100Hz.

3 Recognition

3.1 Acceleration Data Analysis

Figure 2 (left) shows a segment of the acceleration data collected during the experiment. The segment includes sawing, removing the wood from the vise, and drilling. The user accesses the drawer two times and walks between the vise and the drill. Clear differences can be seen in the acceleration signals. For example, sawing clearly reflects a periodic motion. By contrast, the drawer access (marked as 1a and 1b in the figure) shows a low frequency "bump" in acceleration. This bump corresponds to the 90 degree turns of the wrist as the user releases the drawer handle, retrieves the object, and grasps the handle again to close the drawer.

Given the data, time series recognition techniques such as hidden Markov models (HMMs) [13] should allow the recognition of the relevant gestures. However, a closer analysis reveals two potential problems. First, not all relevant activities are strictly constrained to a particular sequence of motions. While the characteristic motions associated with sawing or hammering are distinct, there is high variation in drawer manipulation and grinding. Secondly, the activities are separated by sequences of user motions unrelated to the task (e.g the user scratching his head). Such motions may be confused with the relevant activities. We define a “noise” class to handle these unrelated gestures.

3.2 Sound Data Analysis

Considering that most gestures relevant for the assembly/maintenance scenario are associated with distinct sounds, sound analysis should help to address the problems described above. We distinguish between three different types of sound:

1. *Sounds made by a hand-tool:* - Such sounds are directly correlated with user hand motion. Examples are sawing, hammering, filing, and sanding. These actions are generally repetitive, quasi–stationary sounds (i.e. relatively constant over time - such that each time slice on a sample would produce an identical spectrum over a reasonable length of time). In addition these sounds are much louder than the background noise (dominant) and are likely to be much louder at the microphone on the user’s hand than on his chest. For example, the intensity curve for sanding (see Figure 2 top right) reflects the periodic sanding motion with the minima corresponding to the changes in direction and the maxima coinciding with the maximum sanding speed in the middle of the motion. Since the user’s hand is directly on the source of the sound the intensity difference is large. For other activities it is smaller, however in most cases still detectable.
2. *Semi-autonomous sounds:* These sounds are initiated by user’s hand, possibly (but not necessarily) remaining close to the source for most of the sound duration. This class includes sound produced by a machine, such as the drill or grinder. Although ideal quasi-stationary sounds, sounds in this class may not necessarily be dominant and tend to have a less distinct intensity difference between the hand and the chest (for example, when a user moves their hand away from the machine during operation).
3. *Autonomous sounds:* These are sounds generated by activities not driven by the user’s hands (e.g loud background noises or the user speaking).

Obviously the vast majority of relevant actions in assembly and maintenance are associated with handtool sounds and semi–autonomous sounds. In principle, these sounds should be easy to identify using intensity differences between the wrist and the chest microphone. In addition, if extracted appropriately, these sounds may be treated as quasi-stationary and can be reliably classified using simple spectrum pattern matching techniques.

The main problem with this approach is that many irrelevant actions are also likely to fall within the definition of hand-tool and semi–autonomous sound.

Such actions include scratching or putting down an object. Thus, like acceleration analysis, sound–based classification also has problem distinguishing relevant from irrelevant actions and will produce a number of false positives.

3.3 Recognition Methodology

Neither acceleration nor sound provide enough information for perfect extraction and classification of all relevant activities; however, we hypothesize that their sources of error are likely to be statistically distinct. Thus, we develop a technique based on the fusion of both methods. Our procedure consists of three steps:

1. Extraction of the relevant data segments using the intensity difference between the wrist and the chest microphone. We expect that this technique will segment the data stream into individual actions
2. Independent classification of the actions based on sound or acceleration. This step will yield imperfect recognition results by both the sound and acceleration subsystems.
3. Removal of false positives. While the sound and acceleration subsystems are each imperfect, when their classifications of a segment agree, the result may be more reliable (if the sources of error are statistically distinct).

4 Isolated Activity Recognition

As an initial experiment, we segment the activities in the data files by hand and test the accuracy of the sound and acceleration methods separately. For this experiment, the non-tool gestures, drawer and clapping, are treated as noise and as such are not considered here.

4.1 Accelerometer–Based Activity Recognition

Hidden Markov models (HMMs) are probabilistic models used to represent non-deterministic processes in partially observable domains and are defined over a set of states, transitions, and observations. Details of HMMs and the respective algorithms are beyond the scope of this paper but may be found in Rabiner's tutorial on the subject [13].

Hidden Markov models have been shown to be robust for representation and recognition of speech [9], handwriting [19], and gestures [21]. HMMs are capable of modeling important properties of gestures such as time variance (the same gesture can be repeated at varying speeds) and repetition (a gesture which contains a motion which can be repeated any number of times). They also handle noise due to sensors and imperfect training data by providing a probabilistic framework.

For gesture recognition, a model is trained for each of the gestures to be recognized. In our experiment, the set of gestures includes saw, drill, screw, hammer, sand, file and vise. Once the models are trained, a sequence of features can be passed to a recognizer which calculates the probability of each model given the observation sequence and returns the most likely gesture. For our experiments, the set of features consist of readings from the accelerometers positioned at the wrist and at the elbow. This provides 6 total continuous feature values - the x,y and z acceleration readings for both positions - which are then normalized to sum to one and collected at approximately 93 Hz.

We found that most of the workshop activities typically require only simple single Gaussian HMMs for modeling. For file, sand, saw, and screw, a 5 state model with 1 skip transition and 1 loop-back transition suffice because they consist of simple repetitive motions. Drill is better represented using a 7 state model, while grinding is again more complex, requiring a 9 state model. The vise is unique in that it has two separate motions, opening and closing. Thus a 9 state model is used with two appropriate loop-backs to correctly represent the gesture (See Figure 3). These models were selected through inspection of the data, an understanding of nature of the activities, and experience with HMMs.

4.2 HMM Isolation Results

For this project, a prototype of the Georgia Tech Gesture Recognition Toolkit was used to train the HMMs and for recognition. The Toolkit is an interface to the HTK toolkit [26] designed for training HMMs for speech recognition. HTK handles the algorithms for training and recognizing the Hidden Markov Models allowing us to focus primarily on properly modeling the data.

To test the performance of the HMMs in isolation, the shop accelerometer data was partitioned by hand into individual examples of gestures. Accuracy of the system was calculated by performing leave-one-out validation by iteratively reserving one sample for testing and training on the remaining samples for each sample. The HMMs were able to correctly classify 95.51% of the gestures over data collected from the shop experiments. The rates for individual gestures are given in Table 2.

4.3 Sound Recognition

Method

The basic sound classification scheme operates on individual frames of length t_w seconds. The approach follows a three step process: feature extraction, dimensionality reduction, and the actual classification.

The features used are the spectral components of each t_w obtained by Fast Fourier Transformation (FFT). This produces $N = \frac{f_s}{2} \cdot t_w$ dimensional feature vectors, where f_s is sample frequency. Rather than attempting to classify such large N-dimensional vectors directly, Linear Discriminant Analysis (LDA)[6] is employed to derive an optimal projection of the data into a smaller, M dimensional feature space (where M is the number of classes). In the "recognition phase", the LDA transformation is applied to the data frame under test to produce the corresponding $M - 1$ dimensional feature vector.

Using a labeled training-set, class means are calculated in the $M-1$ dimensional space. Classification is performed simply by choosing the class mean which has the minimum Euclidean distance from the test feature vector (see Figure 2 bottom right).

Intensity Analysis

Making use of the fact that signal intensity is inversely proportional to the square of the distance from its source, the ratio of the two intensities I_{wrist}/I_{chest} is used as a measure of absolute distance of source from the user. Assuming the sound source is distance d from the wrist microphone and $d+\delta$ from the chest, the ratio of the intensities will be proportional to

$$\frac{I_{wrist}}{I_{chest}} \simeq \frac{(d+\delta)^2}{d^2} = \frac{d^2+2d\delta+\delta^2}{d^2} = 1 + \frac{2\delta}{d} + \frac{\delta^2}{d^2}$$

When both microphones are separated by at least δ, any sound produced at a distance d (where $d >> \delta$) from the user will bring this ratio close to one. Sounds produced near the chest microphone (e.g. the user speaking) will cause the ratio to approach zero whereas any sounds close to the wrist mic will make this ratio large.

Sound extraction is performed by sliding a window w_{ia} over the f_s Hz resampled audio data. On each iteration, the signal energy over w_{ia} for each channel is calculated. For these windows, the difference in ratio I_{wrist}/I_{chest} and its reciprocal are obtained, which are then compared to an empirically obtained threshold th_{ia}.

The difference $I_{wrist}/I_{chest} - I_{chest}/I_{wrist}$ provides a convenient metric for thresholding - zero indicates a far off (or exactly equidistant) sound; while above or below zero indicate a sound closer to the wrist or chest microphone respectively.

4.4 Results

In order to analyze the performance of the sound classification, individual examples of each class were hand partitioned from each of the 10 experiments. This provided at least 10 samples of every class - some classes had more samples on account of more frequent useage (e.g. vise). From these, two samples of each class were used for training while testing was performed on the rest.

Similar work[18] used FFT parameters of f_s=4.8kHz and t_w=50 ms (256 points), for this experiment t_w was increased to 100 ms. With these parameters LDA classification was applied to successive t_w frames within each of the class partitioned samples - returning a hard classification for each frame. Judging accuracy by the number of correctly matching frames over the total number of frames in each sample, an overall recognition rate of 90.18% was obtained. Individual class results are shown in the first column of Table 2. We then used intensity analysis to select those frames corresponding to where source intensity ratio difference surpassed a given threshold. With LDA classification applied only to these selected frames, the recognition improved slightly to a rate of 92.21% (second column of Table 2.)

To make a comparison with the isolated accelerometer results, a majority decision was taken over all individual frame results within each sample to produce an overall classification for that gesture. This technique resulted in 100% recognition over the sound test data in isolation.

Table 2. Isolated recognition accuracy (in %) for sound LDA, LDA with IA preselection, majority decision over IA+LDA, and for acceleration based HMM

	Sound			Acceleration
Gesture	LDA	IA+LDA	maj(IA+LDA)	HMM
Hammer	96.79	98.85	100	100
Saw	92.71	92.98	100	100
Filing	69.68	81.43	100	100
Drilling	99.59	99.35	100	100
Sanding	93.66	92.87	100	88.89
Grinding	97.77	97.75	100	88.89
Screwing	91.17	93.29	100	100
Vise	80.10	81.14	100	92.30
Overall	90.18	92.21	100	95.51

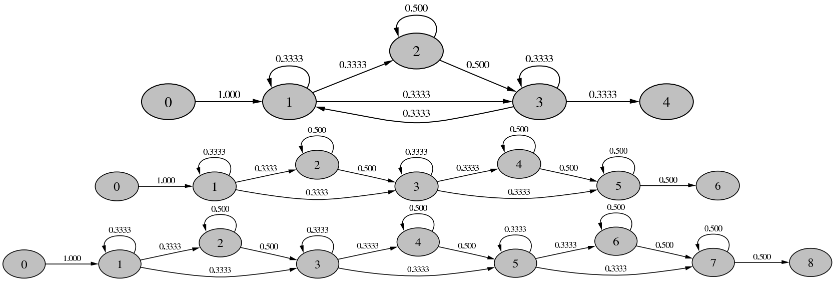

Fig. 3. HMMs topologies

5 Continuous Recognition

Recognition of gestures from a continuous stream of features is difficult. However, we can simplify the problem by partitioning the continuous stream into segments and attacking the problem as isolated recognition. This approach requires a method of determining a proper partitioning of the continuous stream. We take advantage of the intensity analysis described in the previous section as a technique for identifying appropriate segments for recognition.

Since neither LDA nor the HMM are perfect at recognition, and each is able to recognize a different set of gestures well due to working in different feature space, it is advantageous to compare their independent classifications of a segment. If the classification of the segment by the HMMs matches the classification of the segment by the LDA, the classification can be believed. Otherwise, the noise class can be assumed, or perhaps a decision appropriate to the task can be taken (such as requesting additional information from the user).

Thus, the recognition is performed in three main stages: 1) Extracting potentially interesting partitions from the continuous sequence, 2) Classifying these individually using the LDA and HMMs, and 3) Combining the results from these approaches.

5.1 LDA for Partitioning

For classification, partitioned data needs to be arranged in continuous sections corresponding to a single user activity. Such partitioning of the data is obtained in two steps: First, LDA classification is run on segments of data chosen by the IA. Those segments not chosen by intensity analysis are returned with classification zero. (In this experiment, classifications are returned at the same rate as accelerometer features); Secondly, these small window classifications are further processed by a larger (several seconds) majority decision window, which returns a single result for the entire window duration.

This partitioning mechanism helps reduce the complexity of continuous recognition. It will not give accurate bounds on the beginning and end of a gesture. Instead, the goal is to provide enough information to generate context at a general level, i.e., “The user is hammering” as opposed to “A hammering gesture occurred between sample 1500 and 2300.” The system is tolerant of, and does not require, perfect alignment between the partitions and the actual gesture. The example alignment shown in Figure 4 is acceptable for our purposes.

5.2 Partitioning Results

Analysis of the data was performed to test the system’s ability to reconstruct the sequence of gestures in the shop experiments based on the partitioning and recognition techniques described to this point. Figure 5 shows an example of the automated partitioning versus the actual events. The LDA classification of each partition is also shown. For this analysis of the system, the non-tool gestures, drawer and clapping, were considered as part of the noise class. After applying the partition scheme, a typical shop experiment resulted in 25-30 different partitions.

5.3 HMM Classification

Once the partitions are created by the LDA method, they are passed to set of HMMs for further classification. For this experiment, the HMMs are trained on individual gestures from the shop experiments using 6 accelerometer features from the wrist and elbow. Ideally, the HMMs will return a single gesture classification for each segment. However, the segment sometimes includes the beginning or end of the next or previous gesture respectively, causing the HMMs to return a sequence of gestures. In such cases, the gesture which makes up the majority of the segment is used as the classification. For example the segment labeled "B" in Figure 4 may return the sequence "hammer vise" and would then be assigned as the single gesture "vise."

5.4 Combining LDA and HMM Classification

For each partitioned segment, the classification of the LDA and HMM methods were compared. If the classifications matched, that classification was assigned the segment. Otherwise, the noise class was returned.

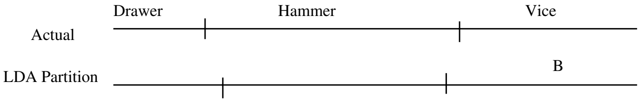

Fig. 4. Detailed example of LDA partitioning

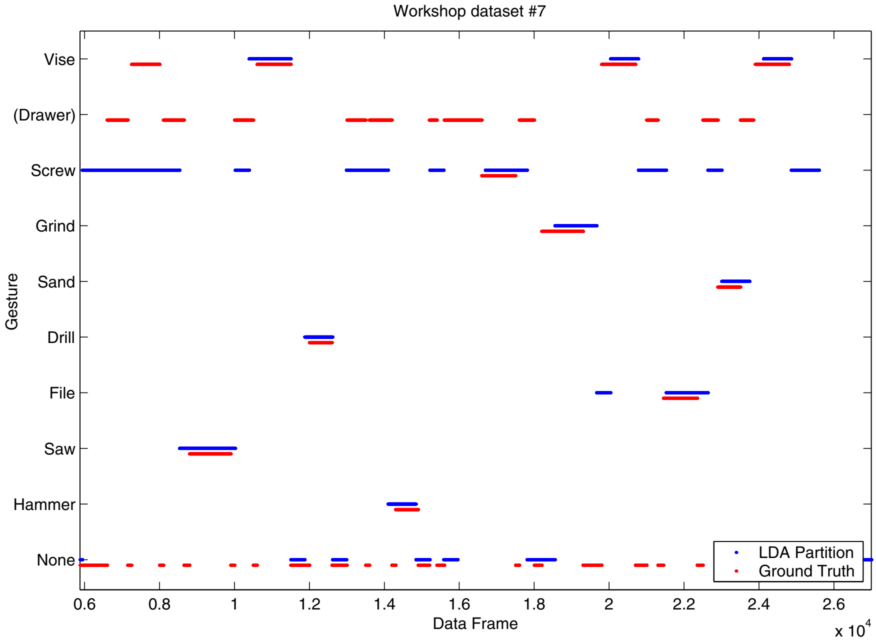

Fig. 5. LDA partitions versus ground truth on a typical continuous dataset

Table 3. Continuous recognition accuracy per gesture (Correct | Insertions | Deletions | Substitutions | Accuracy) and probability of gesture given classification P(G|Class)

Gesture	HMM					LDA					HMM + LDA					
	C	I	D	S	Acc	C	I	D	S	Acc	C	I	D	S	Acc	P(G|Class)
Hammer	8	2	0	1	66.7	9	1	0	0	88.9	8	0	1	0	88.9	1.00
Saw	9	0	0	0	100	9	1	0	0	88.9	9	0	0	0	100	1.00
Filing	10	0	0	0	100	9	7	0	1	23.2	9	0	1	0	90	1.00
Drilling	9	7	0	0	22.2	9	1	0	0	88.9	9	0	0	0	100	1.00
Sanding	8	0	0	1	77.8	9	8	0	0	11.1	8	0	1	0	88.9	1.00
Grinding	11	13	0	0	-18.2	9	0	0	2	81.8	9	0	2	0	81.8	1.00
Screw	5	1	0	4	44.4	9	75	0	0	-733.3	4	0	5	0	44.4	1.00
Vise	42	0	0	1	97.7	34	1	2	7	76.6	36	0	7	0	83.7	1.00
Overall	102	23	0	7	72.5	97	94	2	10	2.8	92	0	17	0	84.4	1.00

Table 3 shows the number of correct classifications (C), insertions (I), deletions (D), and substitutions(S) for the HMMs, the LDA, and the combination. Insertions are defined as noise gestures identified as a tool gesture. Deletions are tool gestures recognized as noise gestures. A substitution for a gesture occurs when that gesture is incorrectly identified as a different gesture. In addition, the accuracy of the system is calculated based on the following metric:

$$\%Accuracy \;=\; \frac{Correct - Insertions}{TotalSamples}$$

The final column reports the probability of a gesture having occurred given that the system reported that gesture.

Clearly, the HMMs and LDA each perform better than the other on various gestures and tended to err in favor of a particular gesture. When incorrect, LDA tended to report the "screw" gesture. Similarly, the HMMs tended to report "grinding" or "drilling." Comparing the classification significantly helps address this problem and reduce the number of false positives, thus increasing the performance of the system as a whole. The data shows that the comparison method performed better than the HMMs and the LDA in many cases and improved the accuracy of the system.

6 Discussion

Although the accuracy of the system in general is not perfect, it is important to note that the combined HMM + LDA method results in no insertions or substitutions. This result implies that when the system returns a gesture, that gesture did occur. While the system still misses some gestures, the fact that it does not return false positives allows a user interface designer to be more confident in his use of positive context.

Of course for many applications deletions are just as undesirable as false positives. In a safety monitoring scenario for example, any deletions of alarm or warning events would naturally be unnaceptable. In such cases it would be better for the system to return some warning, however erroneous, rather than none at all. On the other hand, if one sensor is known to produce many false positives in particular circumstances, whereas another is known to be extremely reliable for the same, then some means of damping the influence of the first in favour of the second sensor would be desirable.

The simple fusion scheme described in this paper could be modified to accomodate these issues by weighting sensor inputs based on knowledge of their reliability in given circumstances. Such weighting, together with decision likelihood information from individual classifiers, would allow a more intelligent fusion scheme to be developed. This will be the focus of future work.

7 Conclusion

We have shown a system capable of segmenting and recognizing typical user gestures in a workshop environment. The system uses wrist and chest worn microphones and accelerometers, leveraging the feature attributes of each modality to improve the system's performance. For the limited set analyzed, the system demonstrated perfect performance in isolated gesture testing and a zero false positive rate in the continuous case. In the future, we hope to apply these promising techniques, together with more advanced methods for sensor fusion, to the problem of recognizing everyday gestures in more general scenarios.

References

1. D. Abowd, A. K. Dey, R. Orr, and J. Brotherton. Context-awareness in wearable and ubiquitous computing. *Virtual Reality*, 3(3):200–211, 1998.
2. Len Bass, Dan Siewiorek, Asim Smailagic, and John Stivoric. On site wearable computer system. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 2 of *Interactive Experience*, pages 83–84, 1995.
3. Michael C. Büchler. *Algorithms for Sound Classification in Hearing Instruments.* PhD thesis, ETH Zurich, 2002.
4. C. Buergy, Jr. J. H. Garrett, M. Klausner, J. Anlauf, and G. Nobis. Speech-controlled wearable computers for automotive shop workers. In *Proceedings of SAE 2001 World Congress*, number 2001-01-0606 in SAE Technical Paper Series, march 2001.
5. B. Clarkson, N. Sawhney, and A. Pentland. Auditory context awareness in wearable computing. In *Workshop on Perceptual User Interfaces*, November 1998.
6. R. Duda, P. Hart, and D. Stork. *Pattern Classification, Second Edition.* Wiley, 2001.
7. J. H. Garrett and A. Smailagic. Wearable computers for field inspectors: Delivering data and knowledge-based support in the field. *Lecture Notes in Computer Science*, 1454:146–164, 1998.
8. N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Tröster. Wearable sensing to annotate meeting recordings. In *6th Int'l Symposium on Wearable Computers*, pages 186–193, October 2002.

9. F. Kubala, A. Anastasakos, J. Makhoul, L. Nguyen, R. Schwartz, and G. Zavaliagkos. Comparative experiments on large vocabulary speech recognition. In *ICASSP*, Adelaide, Australia, 1994.
10. Paul Lukowicz, Holger Junker, Mathias Staeger, Thomas von Bueren, and Gerhard Troester. WearNET: A distributed multi-sensor system for context aware wearables. In G. Borriello and L.E. Holmquist, editors, *UbiComp 2002: Proceedings of the 4th International Conference on Ubiquitous Computing*, pages 361–370. Springer: Lecture Notes in Computer Science, September 2002.
11. J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with multiple acceleration sensors. In *2001 IEEE International Conference on Systems, Man and Cybernetics*, volume 3494, pages 747–752, 2001.
12. 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.
13. L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–16, January 1986.
14. C. Randell and H. Muller. Context awareness by analysing accelerometer data. In *Digest of Papers. Fourth International Symposium on Wearable Computers.*, pages 175–176, 2000.
15. J. M. Rehg and T. Kanade. DigitEyes: vision-based human hand tracking. School of Computer Science Technical Report CMU-CS-93-220, Carnegie Mellon University, December 1993.
16. J. Schlenzig, E. Hunter, and R. Jain. Recursive identification of gesture inputs using hidden Markov models. *Proc. Second Annual Conference on Applications of Computer Vision*, pages 187–194, December 1994.
17. D. Sims. New realities in aircraft design and manufacture. *Computer Graphics and Applications*, 14(2), March 1994.
18. Mathias Stäger, Paul Lukowicz, Niroshan Perera, Thomas von Büren, Gerhard Tröster, and Thad Starner. Soundbutton: Design of a low power wearable audio classification system. 7th Int'l Symposium on Wearable Computers, 2003.
19. T. Starner, J. Makhoul, R. Schwartz, and G. Chou. On-line cursive handwriting recognition using speech recognition methods. In *ICASSP*, pages 125–128, 1994.
20. T. Starner, B. Schiele, and A. Pentland. Visual contextual awareness in wearable computing. In *IEEE Intl. Symp. on Wearable Computers*, pages 50–57, Pittsburgh, PA, 1998.
21. T. Starner, J. Weaver, and A. Pentland. Real-time American Sign Language recognition using desk and wearable computer-based video. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 20(12), December 1998.
22. Bernt Schiele Stavros Antifakos, Florian Michahelles. Proactive instructions for furniture assembly. In *4th Intl. Symp. on Ubiquitous Computing. UbiComp 2002.*, page 351, Göteborg, Sweden, 2002.
23. K. Van-Laerhoven and O. Cakmakci. What shall we teach our pants? In *Digest of Papers. Fourth International Symposium on Wearable Computers.*, pages 77–83, 2000.
24. C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *ICCV*, Bombay, 1998.
25. 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.
26. S. Young. *HTK: Hidden Markov Model Toolkit V1.5.* Cambridge Univ. Eng. Dept. Speech Group and Entropic Research Lab. Inc., Washington DC, 1993.