Gesture Spotting for Controlling a Mobile Assistance System for Service and Maintenance

Albert Hein[1] , Thomas Low[2], Maximilian Hensch[1], Thomas Kirste[1], Andreas Nürnberger[2]

[1] Dept. of Computer Science, University of Rostock, Germany
{albert.hein, maximilian.hensch, thomas.kirste}@uni-rostock.de
[2] Data and Knowledge Engineering Group, Otto-von-Guericke University Magdeburg, Germany
{thomas.low, andreas.nuernberger}@ovgu.de

Abstract: As information technology interweaves with our daily environment, new modes of interaction will be required. In this paper, we suggest a gesture-based approach and present a prototypical case study for a gesture controlled mobile assistant system for an industrial application domain. We discuss a number of design requirements, that are common to gesture-based interaction techniques, describe the resulting system architecture, as well as the training and recognition process, and an experimental evaluation. We will show that even without too much effort such a system can be realized with standard hardware and algorithms, and still achieve very robust recognition results.

1 Introduction

More and more computing power is becoming available in devices small enough to hide and sometimes even get invisible in our environment. This paradigm shift is often referred to as *ubiquitous computing* or *ambient intelligence*, and also emerges in the field of *wearable computing*. While context and intention recognition are widely seen as a major prerequisite, direct and explicit interaction between the human and the technology will still be needed. But how to interact with disappearing technology and distributed devices?

Starting from a rough industrial environment, in this paper we present one possible approach to human computer interaction using free handed gesture recognition. Our prototype is capable of robustly controlling a mobile assistance system on a tablet computer with a sensor watch and without the need of explicit start and stop trigger events. First, we give a short insight into related work covering gesture recognition, gesture spotting as well as activity spotting. The remaining part of the paper will discuss the case study, its design requirements, followed by a detailed description of the system architecture, the training and recognition process, as well as the experimental evaluation.

2 Related Work

Gestures are one of the fundamentals of non-verbal communication and the idea of utilizing them for human computer interaction gained a lot of interest among researchers during the last years. While the first publications applied computer vision methods, this approach is not always suitable as it is dependent on camera infrastructure, adequate lighting and high computing capacity. When small and low cost acceleration sensors became widely available and integrated into mobile devices, the detection of free handed gestures got into the focus again. First approaches were implemented using custom built sensing boards [FABR05], later work is done mostly on hand-held mobile devices [NH08, PHGP08]. The Wii Controller showed the potential of gesture recognition for HCI and the researchers creativity when becoming commercially available [SPHB08]. However, being dependent on holding a piece of hardware in the hand, these devices do not allow hands-free work, which is why we and others developed a bracelet prototype or – when finally available – utilized smart watches [Alb09, Ger11, Oli09].

Most of these approaches are built upon *Hidden Markov Models* (HMMs) [SPHB08, KKM$^+$06, Oli09, Tho08], but also *Dynamic Time Warping* [Jia08], *Decision Trees* [Alb09] or a mixture of multiple classifiers [Dav07, Zol08], which achieve high recognition accuracies. Unfortunately, many of the latter are not able to handle complex gestures, multiple users without individual training, or need many annotated training examples. HMMs on the other hand are able to handle complex temporal relations, offer a clean probabilistic foundation, and are widely used and well understood. The generative nature of HMMs allows for manual modelling and interpretation and model parameters can be learned automatically. As the computational power of mobile devices continuously increases, it is already possible to evaluate multiple HMMs in parallel in real time directly on the device.

However, although the classification task can be regarded as solved, a remaining key problem is to identify meaningful gestures in a continuous stream of sensor data. This is referred to as *gesture spotting* and is still in its first steps. Many ideas for this are taken from the more general field of *activity spotting* [LO98, BS09, Amf11]. In our case study, the fact that there is only one sensor available at the wrist, makes it difficult to find the correct start and end point of a gesture. In fact [Jam05] claims, that it is even impossible to separate gestures from non-gestures without other sensing modalities like sound. In [Ger11] gesture spotting is realised using a special *double click* gesture to trigger the beginning and the end of a so called interaction phase. In [Oli09] the term spotting is reduced to simply identify well defined gestures where otherwise the hand is idle and no motion occurs at all. Pressing a button like in [Jia08] cannot be accepted. Recent exponential smoothing based segmentation methods [Zol08, Tho08] are currently most encouraging in our application domain.

3 Case Study

In the following we will discuss a case study where we designed and implemented a gesture spotting algorithm for the non-interrupting interaction with a mobile assistance system. The research project *MAWI*[1] (Mobile 3D-Assistance for Service and Maintenance) aims at providing a worker with context dependent documentation on a mobile device (Android tablet). The assistant is intended to be deployed in shipyards, engine rooms or other industrial facilities where technicians maintain and repair machines on site. Work gloves, dirt and the need for free hands do not allow for directly operating a touch device. Speech recognition is inapplicable due to background noise and optical methods fail because of inappropriate lighting conditions. As the workflow should be interrupted as minimal as possible, standard human computer interaction methods fail in this scenario. Free handed gestures appear to be an acceptable way of interaction as they require no specialized infrastructure and no cumbersome disruptions. The aim of the following algorithm is not to achieve the highest possible accuracy for gesture recognition in literature but to provide a straight forward and robust example on how to design such an interaction modality. Following this idea we only utilized standard machine learning algorithms and off the shelf hardware components.

We will not focus on the Assistance System itself here, as we are only interested in controlling the end user front end. Its components are an Android tablet based documentation assistant for the end user (=*MAWI Client*), an authoring tool for the creation of the technical documentation, and a semantic expert system as the knowledge base for storing the content. The client software is controllable via Android Intents, a native IPC API and expects commands like *next page*, *previous page*, *scrolling*, *confirm*, or *abort*.

The selection of prototypical control gestures is taken from the literature [Ger11, Alb09] and adapted to the specific use case to improve uniqueness and avoid ambiguities regarding non-gestures during the normal work flow. Other criteria were subjective familiarity with these gestures in the daily life for increasing the intuitivity and a short duration to support responsiveness. The final seven gestures considered in the tests are *screw*, *circle*, *double wipe*, *move up*, *turn over (page)*, *double tap*, and *fist bump*.

3.1 Design Requirements

We identified the following design requirements for this case study, which we believe have to be met to gain usefulness and the users' acceptance.

Unobtrusive Controlling the assistance system should not disturb the operative workflow. In this study, we apply *gesture spotting*, which allows to autonomously differentiate gestures and non-gestures without an explicit start trigger event.

Ergonomic The sensor equipment must neither constrain the mobility of the worker's hands, nor being perceived as limiting. We use a wireless wrist watch that contains

[1] http://www.wartungsassistenz.de

the required senors. It is small and lightweight, and easy to attach without the help of another person.

Inexpensive, Rough, Mobile The wrist watch is low-cost and consist of commercially available standard components, see Section 3.2. It is protected through the worker's gloves from difficulties arising from the field of operation including dirt and pressure. Since the wrist watch is a mobile device, the system architecture was designed such that it can cope with low memory, computing power, and bandwidth.

Reactive Due to the requirements of online interaction, the system was designed to provide a high level of responsiveness. This is achieved through a number of design decision discussed in Section 3.2, e.g. by a preprocessing and segmentation of the raw sensor data on the watch.

Intuitive The control gestures resemble intuitive motions used for controlling similar devices to minimize training effort, despite being clearly distinguishable from each other and non-gestures occurring during the typical work flow.

Reliable The reliability of the recognition process is of fundamental importance for user acceptance. We target a recognition accuracy of more than 95%, where the error costs of missed gestures are significantly lower than the costs of erroneously recognized gestures and thereby falsely executed commands on the device. This includes the robustness against inner class variations during gesture execution like temporal and spatial deviations.

User-independent The system is utilizable for multiple persons without the need for separate and individual training. This already requires involving multiple users in the training phase and evaluation of the applied algorithms for the recognition system, see Section 3.4.

Modular The Hidden Markov Models each describe a single gesture and can therefore be replaced and extended easily. Changes do not require a complete re-training of all gestures.

3.2 Hardware and System Architecture

A Texas Instruments MetaWatch (Fig. 1) is used for data acquisition and worn on the dominant wrist of the user. It contains a MSP430F5438A microcontroller (25 MHz, 256 kB Flash, 16 kB RAM) and a KXTF9-1026 3-axis accelerometer[2] capable of measuring $\pm 2g$, $4g$ or $8g$. The device runs a customized FreeRTOS operating system and is communicating over a proprietary Bluetooth stack. Technical firmware problems resulting in an unusable sensor had to be solved in cooperation with Texas Instruments beforehand.

The schematic architecture of the recognition system is very straight forward and is illustrated in Fig. 2. In the presented prototype raw data is read from the sensor and preprocessed on the microcontroller of the watch and then sent to the tablet via Bluetooth. The

[2] http://www.kionix.com

Figure 1: Texas Instruments MetaWatch running the prototype firmware.

preprocessing covers feature extraction and autonomous segmentation (for details see next section). For the evaluation of the algorithm this was still done offline. At the time of writing this has been integrated into the firmware in order to lower the bandwidth needed for the wireless communication and therefore to significantly reduce energy consumption.

Valid data segments are then forwarded to different pretrained HMMs on the tablet which can be loaded dynamically from the software. Each HMM represents one specific gesture. The observation sequence is forward filtered and the loglikelihoods for every HMM are estimated. The spotting step then compares these values and decides via thresholds which sequence belongs to which gesture or whether it origins from random motion.

Inter-process communication on Android can be realized via the native Intents API. Processes can declare themselves as Broadcast Receivers and then may listen to Broadcast Intents they have subscribed to, like telephone ring or battery status. The prototype developed in this case study uses this framework and advertises detected gesture commands as Broadcast Intents to the system. The MAWI Client itself implements a Broadcast Receiver and a corresponding Intent-Filter and assigns a control action to each defined gesture. Reversely, this way it is also possible to backwards communicate with the watch, if needed.

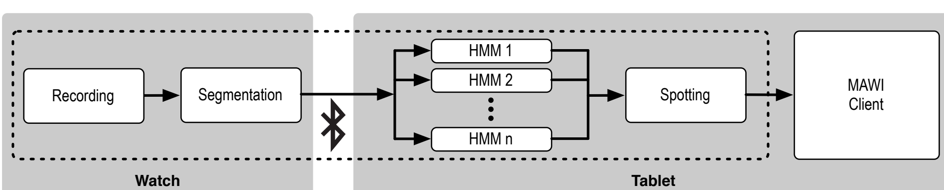

Figure 2: Schematic view of the System Architecture. Gray areas symbolize the two mobile devices watch and tablet. The dashed line contours the components necessary for gesture spotting developed in this case study. Watch and tablet communicate via Bluetooth packets, the MAWI Client on the tablet is listening for controls via Android Intents.

3.3 Training and Recognition Process

After declaring the system setup we now will explain the concepts behind the training and recognition process itself, starting from the sensor data recording and ending with the recognized control gestures.

Recording and Features

According to [Car97], human arm motion happens in a range of approximately $\pm 2.5g$ at an average frequency of $5Hz$ with a maximum of $18Hz$ for single reflexes of which latter ones are not to be expected within the repertory of gestures. Therefore, the actual sampling is done at a rate of $30Hz$ at a range of $\pm 4g$ in 3 dimensions with respect to Nyquist-Shannon.

Preprocessing sensor data usually includes applying a floating window algorithm over the data stream and then calculating significant features like specific frequencies, peaks, zero crossing, statistics or other (see [Alb09]). After a visual analysis of the recorded data, this standard method seemed unnecessary as the majority of the information needed for discrimination is embedded in the temporal structure of the different accelerations. In the following only the raw 3-dimensional acceleration vector $\vec{a}$ for each time step and the norm $|\vec{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$ of this vector is taken into consideration separately for the study. As with the norm all directional information is cropped, mirrored arm movements will not be distinguishable. Still, this one-feature approach already delivers good results in our evaluation.

Segmentation

In order to be independent from a distinguished start signal before beginning the execution of a gesture, a suitable autonomous segmentation algorithm had to be found. In the present case it could be observed that the worker shortly interrupts the momentary motion and intuitively rests his hand for split seconds before starting the gesture and does the same afterwards. When looking at the raw data, gestures start in a resting position, followed by rapid acceleration, going over to continuous direction changes, and ending in an idle state again. This can be captured using an exponential smoothing. Similar approaches can be found in [Fra98] and [Zol08], although they expect the hand to rest while not executing a gesture. According to our gesture definition, first the acceleration change for each timestep is determined. As the recording is discrete, this derivation simply is the difference between two consecutive measurements. Since we are only interested in the euclidean distance, the following formula is used:

$$H_k = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2 + (z_k - z_{k-1})^2}$$

These values are too inconsistent to be considered directly for segmentation. Therefore, an exponential moving average (EMA) is calculated, which puts more weight on the current value than on older ones. This ensures that every new calculation of the EMA is influenced by the preceding ones, whereby the influence decreases the more in the past they are:

$$EMA_{H_k} = \alpha H_k + (1-\alpha)EMA_{H_{k-1}}$$

Test series have shown that for the given sensor configuration an α of 0.3 delivers acceptably smoothed values without averaging too much. In the last step three thresholds are applied to finish the segmentation. First, the incoming sensor data stream is compared to a minimum mean acceleration change of 0.07. If this is reached, time steps are logged until the EMA falls under this threshold again. If this sequence is between 15 and 60 time steps (at $30Hz$ this corresponds to a duration between 0.5 and $2sec$) the segment is considered as a potential gesture. Shorter or longer observations are ignored, avoiding the false detection of short unintended arm movements, weak motion like tremors, and longer continuous activities.

Recognition and Training

For each gesture a separate linear Hidden Markov Model has been trained on a set of prerecorded corresponding data sequences. State specific observation probabilities have been modeled as multi-variate normal distribution (3 dimensional acceleration vector $\vec{a}$) and uni-variate normal distribution (norm of vector $|\vec{a}|$) respectively.

For the utilized Baum-Welch learner feasable initial parameters for the HMM are needed. For the prototype we hand crafted these initial models based on 20 example sequences for each gesture. Number of states, priors, transitions and observation probability parameters have been determined manually. Fig. 3 shows an example for such an initial model. [3] The Baum-Welch algorithm is then executed on a much larger set of training examples and repeated, until the increase in loglikelihood (for explaining the corresponding gesture) falls under a certain ϵ in a training cycle, which guarantees at least locally optimal model parameters. The larger the training set, the more robust the recognition – too small training sets would lead to overfitting. For more details on the training sets used see 3.4.

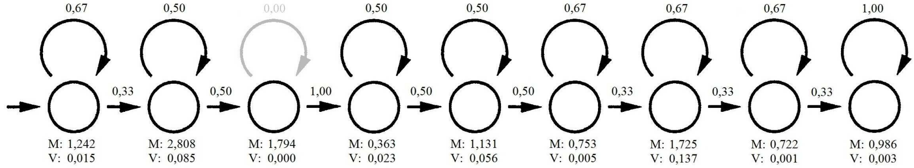

Figure 3: Linear HMM example generated from one observation sequence for the *screw* gesture

All trained models receive the incoming segment of the potential gesture in parallel. Each one separately infers its loglikelihood for the observation sequence using forward backward filtering. These values reflect how well each gesture model explains the recorded

[3]For extending the prototype with new gestures also more autonomous algorithms like *Segmental k-Means* [Bii90] are feasible methods for initialization.

sensor data. In the optimal case the model with the highest loglikelihood matches the gesture. If responsiveness is crucial this approach is not limited to analysing complete sequences, but could already start comparing models when the beginning of a segment was detected using forward filtering.

Spotting

Spotting is the process of autonomously detecting gesture segments in a continuous stream of motion and separating anticipated gestures from random actions. An essential part has already been accomplished during the segmentation, only delivering gesture-like observation sequences with a feasible duration. As mentioned in the section above, the model with the highest loglikelihood indicates the most probable gesture. This method is used in the section 3.4 for the validation, as a test sequence only contains well defined gestures. But to ensure that the sequence does not belong to a phase of random motion, it is important to take the models' uncertainty into consideration.

The idea is to filter out segments, the HMMs are unsure about using a threshold. In the prototype a fixed absolute value (-40) for all gestures has been predetermined, although this method is far from optimal, as loglikelihoods may vary drastically between models and recognition processes (± 35). If the best model fitting exceeds this threshold, the segment will be rated as gesture and following actions will be triggered. This method has potential for improvements: Modelwise thresholds should be able to handle the gesture-wise variability better than one global value. Additionally, a relative threshold indicating how far away the guesses of all models are should be utilized, as in contrast to trained gestures a false gesture often is weakly recognized by more than one model at a time. The spotting process will be evaluated in scenario 3 of the following section.

3.4 Evaluation

The questions we tried to answer in the evaluation are: 1st, what recognition accuracy is achievable with the HMM based approach? 2nd, does the norm of the acceleration vector suffice for a robust recognition and when will directional information be needed? 3rd, how robust is the recognition regarding multiple users without individual training? And 4th, how well does the spotting algorithm perform in a live test over a longer period of time?

Scenario 1: One test person, 10-fold cross validation

For the first test, gestures of one person were recorded. The models have been initialized using 20 examples for each gesture and then ten-fold cross validated on 100 recordings of each of the seven gestures. That makes a total training/test set of 700 recorded gesture executions. The data stream was automatically segmented and then passed to the HMMs. The confusion matrices are presented in Table 1 and have been determined by simply selecting the Model with the highest loglikelihood as no false gestures occurred in the recordings.

screw	circle	double wipe	move up	turn over	double tap	fist bump		screw	circle	double wipe	move up	turn over	double tap	fist bump
100	-	-	-	-	-	-	screw	100	-	-	-	-	-	-
-	100	-	-	-	-	-	circle	-	100	-	-	-	-	-
-	-	100	-	-	-	-	double wipe	-	-	100	-	-	-	-
1	-	-	99	-	-	-	move up	-	-	-	100	-	-	-
-	-	-	-	100	-	-	turn over	-	-	-	-	100	-	-
-	-	-	-	-	100	-	double tap	-	-	-	-	-	100	-
-	-	1	-	-	-	99	fist bump	-	-	-	-	-	-	100

features: $|\vec{a}|$ features: a_x, a_y, a_z

Table 1: Confusion matrices for 10-fold cross validation for one test person. On the left the results for the norm of the acceleration vector are shown, on the right for the plain vector.

The accuracies are 99.71% for the norm and 100.00% in the 3-dimensional case. Regarding the 1 person case these results show, that the HMM-based approach for recognizing this set of gestures is very well suited. Also at least for the selected gestures dropping directional information in the data does not significantly decrease the recognition performance.

Scenario 2: Four test persons, leave one out validation

As the assistance system should be used for multiple users without extra training, it was tested on 4 persons using the same settings. Beforehand for each user 150 executions of each gesture have been recorded. Again, the models have been initialized using 20 examples for each gesture and were then tested on the new 4200 observation sequences using leave one person out. The results are shown in Table 2.

screw	circle	double wipe	move up	turn over	double tap	fist bump		screw	circle	double wipe	move up	turn over	double tap	fist bump
599	-	-	-	-	1	-	screw	600	-	-	-	-	-	-
9	587	-	2	2	-	-	circle	1	582	-	4	-	12	1
1	1	597	-	-	1	11	double wipe	-	-	592	-	-	8	-
16	7	-	577	-	-	-	move up	-	-	-	600	-	-	-
10	-	-	-	590	-	-	turn over	9	-	-	-	591	-	-
2	-	-	-	-	597	19	double tap	-	-	-	-	-	600	-
2	1	8	-	-	9	580	fist bump	-	-	-	-	-	-	600

features: $|\vec{a}|$ features: a_x, a_y, a_z

Table 2: Confusion matrices for leave-one-out validation for four test persons. Again, on the left the results for the norm of the acceleration vector are shown, on the right for the plain vector.

The accuracies are 97.83% for the norm and 99.16% in the 3-dimensional case. These results are surprisingly good, especially in the case where only the overall acceleration is considered. It could be shown, that no individual training per user is needed to achieve a very accurate and robust recognition.

Scenario 3: One test person, continuous gesture spotting

Last, the feasibility of the spotting algorithm has been evaluated, which is crucial for the acceptance of the final assistance system by users. In order to simulate a rough environment, we recorded 30 minutes of random arm activity like rotating a control switch, wiping over a table, taking and putting objects which in some cases look very similar to the control gestures. These movements were interrupted from time to time by the seven anticipated gestures where each gesture has been performed 40 times in random order (= 280 in total). Only the 3-dimensional feature set has been tested. The segmentation algorithm already dropped 103 too long and 807 too short sequences, passing a total number of 602 gestures to the HMMs. The HMMs have been trained the same way as before, this time using all 700 recordings from the single person test (same person as now). The absolute global loglikelihood threshold for the spotting was set to -40 (see 3.3 / Spotting). The confusion matrix is shown in Table 3.

	screw	circle	double wipe	move up	turn over	double tap	fist bump
screw	33	4	-	-	-	4	5
circle	-	10	2	-	1	8	8
double wipe	-	-	29	1	-	6	5
move up	1	2	-	10	-	3	2
turn over	-	1	1	-	21	2	7
double tap	1	1	-	-	1	45	5
fist bump	-	3	3	2	-	5	27

features: a_x, a_y, a_z

Table 3: Confusion matrix for the live test

As the test setting was far from optimal, the accuracy of 60.71% is much lower this time, but still by far better than guessing (14.3%). This result appears more promising, when it is taken into consideration, that the 280 gestures only made a small part of the sensor recordings and even after the segmentation had to be identified within 602 potential gesture segments. Unfortunately the confusion matrix is hiding these complications. Also, not all gestures have been spotted: From the 602 potential segments only 259 have been labeled as gestures in the spotting step, which makes a total of 92.5% of the expected 280 performed gestures. While the segmentation algorithm seems to work very reliably, the spotting step needs further investigation like already mentioned before. Still, this initial test gives an outlook on future potentials.

4 Conclusion

While information technology is becoming ubiquitous and is moving away from the desktop to mobile devices getting smaller and smaller, and even further disappearing, the question arises how to intuitively interact with our digital neighbourhood. Free handed gestures may be one unobtrusive possibility to control mobile assistants, even in rough environments, but automatically discriminating gestures from random motion is still in its early stages. In this paper we outlined general requirements which have to be considered when implementing gesture based interaction solutions. We presented a case study of a gesture controlled assistance system for service and maintenance workers which illustrates how such a technique can be realized straight forward without too much effort. Although the achieved recognition rate is very high, there is a lot of potential for future optimizations, especially regarding the ad-hoc segmentation and spotting algorithm. Also, devices like smart watches offer many opportunities and should be seen in an even more holistic way: Apart from being a cheap and unobtrusive input device they also offer tactile (vibration), visual (display), and acoustic feedback and open a complete new field of Human Computer Interaction in a world where information technology more and more gets invisible.

Acknowledgements

This work has been supported by FORTecH Software, Rostock.

References

[Alb09] Albert Hein, André Hoffmeyer, Thomas Kirste. Utilizing an Accelerometric Bracelet for Ubiquitous Gesture-based Interaction. In *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, pages 19–24, 2009.

[Amf11] Oliver Amft. Self-Taught Learning for Activity Spotting in On-body Motion Sensor Data. In *Proceedings of the 2011 15th Annual International Symposium on Wearable Computers*, ISWC '11, pages 83–86, Washington, DC, USA, 2011. IEEE Computer Society.

[Bii90] Biing-Hwang Juang, Lawrence R. Rabiner. The Segmental k-Means Algorithm for Estimating Parameters of Hidden Markov Models. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38:1639–1641, 1990.

[BS09] Ulf Blanke and Bernt Schiele. Daily Routine Recognition through Activity Spotting. In *Proceedings of the 4th International Symposium on Location and Context Awareness*, 2009.

[Car97] Carlijn V. C. Bouten, Karel T. M. Koekkoek, Maarten Verduin, Rens Kodde, Jan D. Janssen. A Triaxial Accelerometer and Portable Data Processing Unit for the Assessment of Daily Physical Activity. In *IEEE Transactions on Biomedical Engineering*, pages 136–147, 1997.

[Dav07] David Bannach, Oliver Amft, Kai S. Kunze, Ernst. A. Heinz, Gerhard Tröster, Paul Lukowicz. Waving Real Hand Gestures Recorded by Wearable Motion Sensors to a

Virtual Car and Driver in a Mixed-Reality Parking Game. In *Proceedings of the 2nd IEEE Symposium on Computational Intelligence and Games*, pages 32–39, 2007.

[FABR05] E. Farella, A. Acquaviva, L. Benini, and B. Riccò. A wearable gesture recognition system for natural navigation interfaces. In *Proceedings of EUROMEDIA2005*, pages 110–115, Toulouse, April 2005. D.E.I.S. - UNIVERSITA' DI BOLOGNA.

[Fra98] Frank Hofmann, Peter Heyer, Günter Hommel. Velocity profile based recognition of dynamic gestures with discrete Hidden Markov Models. In *Gesture and Sign Language in Human-Computer Interaction*, pages 81–95. Springer Berlin / Heidelberg, 1998.

[Ger11] Gerald Bieber, Marian Haescher, Christian Peter, Mario Aehnelt, Claas Richter, Holger Gohlke. Hands-free Interaction mittels Handgelenksensoren für mobile Assistenzsysteme. In *6. Multimediakongress Wismar 2011: Netzwerk - Forschung - Innovation*, 2011.

[Jam05] Jamie A. Ward, Paul Lukowicz, Gerhard Tröster. Gesture Spotting Using Wrist Worn Microphone and 3-Axis Accelerometer. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pages 99–104, 2005.

[Jia08] Jiayang Liu, Zhen Wang, Lin Zhong, Jehan Wickramasuriya, Venu Vasudevan. uWave: Accelerometer-based Personalized Gesture Recognition and Its Applications. In *Technical Report TR0630-08, Rice University and Motorola Labs*, 2008.

[KKM+06] Juha Kela, Panu Korpipää, Jani Mäntyjärvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and Di Marca. Accelerometer-based gesture control for a design environment. *Personal Ubiquitous Comput.*, 10(5):285–299, 2006.

[LO98] R. H. Liang and M. Ouhyoung. A Real-Time Continuous Gesture Recognition System for Sign Language. In *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, FG '98, pages 558–, Washington, DC, USA, 1998. IEEE Computer Society.

[NH08] G. Niezen and G.P. Hancke. Gesture recognition as ubiquitous input for mobile phones. In *DAP Workshop at UBICOMP 2008*. University of Pretoria, 2008.

[Oli09] Oliver Amft, Roman Amstutz, Asim Smailagic, Dan Siewiorek, Gerhard Tröster. Gesture-Controlled User Input to Complete Questionnaires on Wrist-Worn Watches. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part II: Novel Interaction Methods and Techniques*, 2009.

[PHGP08] Zoltán Prekopcsák, Péter Halácsy, and Csaba Gáspár-Papanek. Design and development of an everyday hand gesture interface. In *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 479–480, New York, NY, USA, 2008. ACM.

[SPHB08] Thomas Schlömer, Benjamin Poppinga, Niels Henze, and Susanne Boll. Gesture recognition with a Wii controller. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 11–14, New York, NY, USA, 2008. ACM.

[Tho08] Thomas Schlömer, Benjamin Poppinga, Niels Henze, Susanne Boll. Gesture Recognition with a Wii Controller. In *Proceedings of the Second International Conference on Tangible and Embedded Interaction (TEI'08), Bonn, Germany*, 2008.

[Zol08] Zoltán Prekopcsák. Proceedings of the 12th International Student Conference on Electrical Engineering. In *Accelerometer Based Real-Time Gesture Recognition*, 2008.