



Airwriting Recognition using Wearable Motion Sensors

Christoph Amma, Dirk Gehrig, Tanja Schultz
Cognitive Systems Lab (CSL), Karlsruhe Institute of Technology, Germany
(christoph.amma,dirk.gehrig,tanja.schultz)@kit.edu

ABSTRACT

In this work we present a wearable input device which enables the user to input text into a computer. The text is written into the air via character gestures, like using an imaginary blackboard. To allow hands-free operation, we designed and implemented a data glove, equipped with three gyroscopes and three accelerometers to measure hand motion. Data is sent wirelessly to the computer via Bluetooth. We use HMMs for character recognition and concatenated character models for word recognition. As features we apply normalized raw sensor signals. Experiments on single character and word recognition are performed to evaluate the end-to-end system. On a character database with 10 writers, we achieve an average writer-dependent character recognition rate of 94.8% and a writer-independent character recognition rate of 81.9%. Based on a small vocabulary of 652 words, we achieve a single-writer word recognition rate of 97.5%, a performance we deem is advisable for many applications. The final system is integrated into an online word recognition demonstration system to showcase its applicability.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: Input Devices and Strategies; I.5.1 [Computing Methodologies]: Models—*Statistical*

General Terms

Design, Algorithms, Human Factors

1. INTRODUCTION

The field of Human-Computer Interaction has a long tradition of exploring and implementing numerous devices and interfaces. However, the requirements and conditions for such interfaces shifted drastically over the past years due to the changing demands of our global and mobile information-based society. Today's electronic devices and digital assistants fit into the smallest pocket, are ubiquitous, always on-

line and accessible. Although everybody savours the small size of hand-helds, the operation of these devices becomes more challenging - only small keys fit on to the device, if at all. This makes complex operations like the input of text very cumbersome. It requires good eye-sight and hand-eye coordination, it keeps hands and eyes busy, and is difficult if the device or the person is moving during operation. Further considering future augmented reality applications with displays integrated in glasses, there is an obvious need for new input devices offering more natural interaction possibilities. Future wearable computer systems will lose many of their advantages, if the interface would still rely on an extra text input device held in hand.

Fortunately, the technology of sensors is advancing significantly, allowing for very small, body-worn, wearable sensors that foster the design of unobtrusive, mobile, and robust interfaces suitable for intuitive wearable computing systems. Traditionally used sensors, which are small and come at very reasonable costs are accelerometers, gyroscopes and magnetometers. Based on the sensor signals, pattern classification techniques can be applied to recognize the performed gestures. Current research on inertial sensor-based gesture recognition concentrates on the recognition of a small set of very simple predefined single gestures. These gestures can then be mapped to certain functions to build gesture based interfaces. Kim et al. [7] use a pen-like accelerometer-based device to recognize single Roman and Hangul characters written in the air.

While predefined gestures are easier to recognize for the machine, it is more burdensome for the users since they have to learn and memorize these gestures. Therefore, we aim at an user-adaptable approach that does not make any assumptions on the used gestures. Furthermore, our approach reaches beyond [7] by introducing a hands-free device to write Roman characters and even whole words in the air. We developed a data glove instead of a pen, which is less obtrusive than an additional device. We implemented an HMM-based recognizer for whole words by concatenating character models allowing complex text input. For the purpose of this study we limited the input character set to capital Roman letters. However, since we apply statistical models, our system can deal with any kind of one-handed gestures.

2. RELATED WORK

The field of gesture recognition has been extensively studied in the past. Two main approaches can be identified, external

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Augmented Human Conference April 2-3, 2010, Megève, France.

Copyright © 2010 ACM 978-1-60558-825-4/10/04 ... \$10.00.

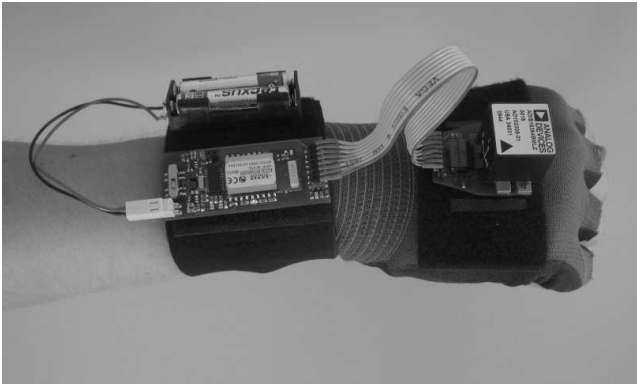


Figure 1: Prototypical CSL data glove.

systems and internal systems, with the latter using body-mounted sensors, including hand-held devices. The former type is traditionally based on non-wearable video based systems and thus is not applicable to our envisioned scenario.

2.1 Sensors and Devices

Different sensing techniques are applied to gesture recognition based on body-worn sensors. Brashear et al. [3] use a hybrid system, consisting of a head mounted camera and accelerometers worn at the wrist for sign language recognition. While such a hybrid system seems promising for sign language recognition, the number and positioning of sensors, particularly those positioned at the head, limit the users' acceptance. Kim et al. [7] use a wireless pen device, equipped with accelerometers and gyroscopes for airwriting character recognition. Their good recognition indicate that airwriting recognition based on accelerometers and gyroscopes is feasible.

Sensors are often integrated into a device, which has to be carried around and held in hand for operation. Therefore, external devices are more obtrusive. Instead, data gloves have been proposed for gesture recognition, for example by Hofmann et al. [6] and Kim and Chien [8]. Usually, these gloves contain a magnetic sensor to determine the global position through an externally induced magnetic field, prohibiting mobile or outdoor applications. These gloves also incorporate many sensors delivering information on finger movement and position, which makes them quite bulky. Alternatively, an accelerometer equipped bracelet for gesture recognition was proposed by Hein et al. [5] and Amft et al. [1], who used accelerometers included in a watch to control certain functions of the watch by gestures. In our work, we combine the advantages of data gloves with the convenience of small hand-held devices by designing and implementing a very slim and unobtrusive interfaces based on a data glove, as depicted in Figure 1. One should be aware that this is a first prototype and that current technology already allows further miniaturization.

2.2 Single Character Classification

Cho et al. [4] and Kim et al. [7] use a pen-type device for single digit and single character recognition using Bayesian Networks. Kim et al. [7] introduce a ligature model based on Bayesian Networks to recognize pen-up and pen-down

strokes and therefore are able to use traditional handwriting training data for their recognizer. Both works rely on the reconstruction of the 3D trajectory. Since errors accumulate quickly over time in this approach, it will only give reasonable results for short periods of time and is thus difficult to apply to the continuous recognition of words. There has also been work on single gesture recognition using a range of other classification methods, such as Naive Bayes, Multilayer Perceptrons, Nearest Neighbour classifiers (Rehm et al. [14]), and SVMs (Wu et al. [15]). Oh et al. [11] use Fisher Discriminant Analysis to recognize single digits. The mentioned methods might perform well on single gesture recognition but are not applicable to recognize a continuous sequence of gestures.

2.3 Continuous Gesture Recognition

Hidden Markov Models have been applied to gesture and handwriting recognition, since they can be easily concatenated to model continuous sequences. A good tutorial introduction in the context of speech recognition was written by Rabiner [13]. Liang et al. [9] make use of HMMs for continuous sign language recognition, but they use desk based Polhemus magnetic sensors. Kim and Chien [8] also used such a sensor to recognize complex gestures based on concatenated HMMs of single strokes. Starner and McGuire [10] use HMMs for continuous mobile american sign language recognition with a 40 word vocabulary, modelling each word by one HMM. Plamondon and Srihari [12] give a survey on traditional on-line and off-line handwriting recognition, where HMMs are applied with success for the on-line case.

It seems that few attempts have been made towards continuous gesture recognition solely with inertial sensors. Furthermore, the set of gestures is also normally limited to a small number of 10 to 20 gestures, which are often chosen and defined for ease of discrimination. In our work, we introduce a system capable to model sequences of primitive gestures and evaluate our system with character gestures.

3. AIRWRITING CHALLENGES

In comparison to conventional online handwriting recognition, typically used on tablet PC's, we have to deal with some specialities when using inertial sensors for writing in the air. In traditional online handwriting recognition, the 2D trajectory of the written strokes is available and classification is done based on this data. In contrast, we get the linear acceleration and the angular velocity of the sensor. Figure 2 shows the raw sensor signals received by writing the character A. The two main differences of our approach from traditional handwriting recognition are the absence of the trajectory information and the pen-up and pen-down movements. The missing pen-up and down movements result in a single continuous stroke for the whole writing, i.e. we get a continuous data stream from the sensors which lacks any segmentation information. If a writer produces several sentences, the result would be one single stroke. This makes the task more difficult, since pen-up and down movements automatically give a segmentation of the data. While this is not a segmentation of characters it gives useful information, we are missing. Also the motion between consecutive characters, which would normally be pen-up, will be represented in the signals we get from the inertial sensors. We face this problem by introducing a repositioning model between the

individual characters of words. The experimental results on word recognition show that this modification of the HMM is suitable.

We also do not get the 3D trajectory easily. While it is theoretically possible to reconstruct the trajectory from a 6 DOF sensor like the one we use by applying a strapdown inertial navigation algorithm, it is practically a hard task because of sensor drift and noise. A standard strapdown algorithm integrates the angular rate once to obtain the attitude of the sensor, then the gravitational acceleration can be subtracted from the acceleration signals and finally double integration of the acceleration yields the position. This triple integration leads to summation of errors caused by sensor drift and noise and after a few seconds the error in position is so high that no reconstruction which is close to the real trajectory is possible. Bang et al. [2] propose an algorithm to reconstruct the trajectory of text written in the air, which works well for isolated characters or short words (in sense of writing time). Since our goal is a system, that allows for arbitrary long input, we avoid this problem by directly working on the sensor signals without estimating the 3D trajectory. The experimental results show that this is feasible in practice.

Gravitation is also a problem, we have to deal with, even without estimating the trajectory. Earth acceleration is always present in the measured acceleration signals and can only be subtracted when the exact attitude of the sensor is known. Due to the mentioned difficulties concerning sensor drift, gravitational acceleration is compensated by subtracting the signal mean. This approximation equals the assumption, that the sensor attitude is constant over the time of one recording. The Variance of the signals is also normalized to reduce the effect of different writing speeds.

4. DATA GLOVE

Our system consists of a thin glove and a wristlet. The glove holds the sensor and the wristlet holds the controller board and the power supply. Figure 1 shows a picture of the system. A microcontroller reads the data from the sensor and sends it over a bluetooth link to a computer, where the recognition is performed. The power supply consists of two rechargeable micro cells allowing operation for more than 4 hours. As sensor, we use an Analog Devices ADIS16364¹ Inertial Measurement Unit. This sensor contains three orthogonal accelerometers for measuring translational acceleration and three orthogonal gyroscopes for measuring angular velocity. The sensor is cubic with an edge length of 23 mm. This is due to the 3-dimensional orientation of the gyroscopes, 3-axis accelerometers are available as standard flat integrated circuits. The measurement range of the accelerometers is -5 g to 5 g with a resolution of 1 mg. The gyroscope measurement range is ± 300 deg/s with a resolution of 0.05 deg/s. The sampling rate is 819.2 Hz. The sensor data is read out by a Texas Instruments MSP430F2132 microcontroller with 8KB Flash memory and 512 Byte RAM, operating at up to 16 MHz. The microcontroller sends the data over an Amber Wireless AMB2300 class 2 Bluetooth Module to a laptop.

¹www.analog.com

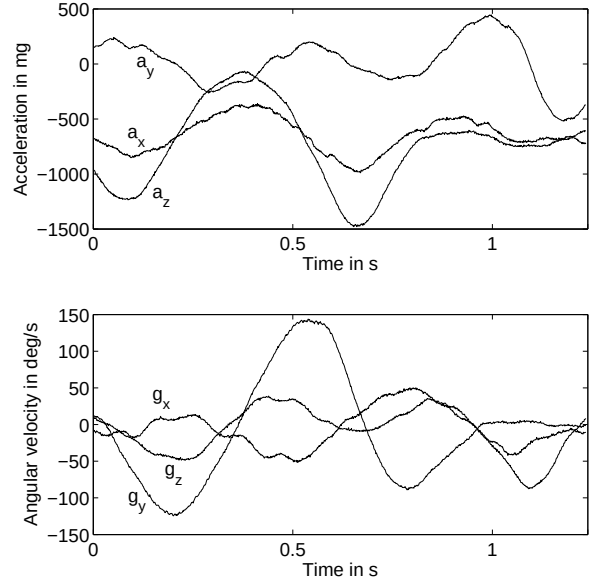


Figure 2: Raw sensor signals for the character A. The upper plot shows accelerometer data, the lower plot shows gyroscope data.

5. DATA ACQUISITION

We have collected character data from 10 subjects, five male and five female. One of the subjects was left-handed, the other nine were right-handers. Every subject wrote 25 times the alphabet resulting in 650 characters per writer and 6500 characters in the dataset in total. In the first recording sessions, the alphabet was recorded in alphabetical ordering, in the latter sessions, every character was recorded in the context of every other. For every subject, data was recorded in one recording session. Word data was collected from one test person which contributed 652 English words to the data set. The words were chosen at random from the list of the “1000 most frequent words in English”² from the corpus of the University of Leipzig. Every word was recorded once. Table 1 gives a summary of the recorded data.

The subjects were sitting on a chair and were asked to write in front of them like writing on an imaginary blackboard. The writing hand was approximately at eye level. Writers were asked to write characters between 10 to 20 cm in height and to keep the wrist fixed while writing. Furthermore, the subjects were told to write in place, which means, the horizontal position should be approximately the same for every character written.

All writing in our experiments was done in capital block letters. Since every writer has its own writing style even for block letters, this led to several writing variants for some letters. These variants are referred to as allographs in handwriting recognition. For example, the letter E has five main allographs even in our small data set. The subjects were asked to be as consistent as possible in the way they write

²<http://www.wortschatz.uni-leipzig.de/html/wliste/html>

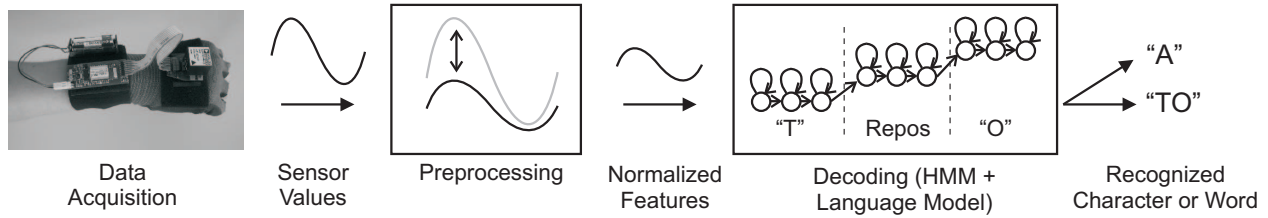


Figure 3: System Overview: Motion data is gathered by the sensors on the glove. The raw data is sent to a computer and preprocessed resulting in a feature vector. The feature vectors are classified using an HMM decoder in combination with a language model. The recognized character or word is the output of the system.

Writer	Char.	Samples	Time
Character Recordings			
A	650	555359	11m 18s
B	650	964185	19m 37s
C	650	733603	14m 56s
D	650	924252	18m 48s
E	650	717878	14m 36s
F	650	682580	13m 53s
G	650	594320	12m 05s
H	650	1122828	22m 51s
I	650	604499	12m 18s
J	650	779136	15m 51s
Word Recordings			
A	3724	2332410	47m 27s

Table 1: Overview of the collected data recordings. The samples column gives the total amount of sensor data samples and the time column the corresponding recording time.

the letters, i.e. stick to one writing variant. When making mistakes, the subjects were able to correct themselves, they were not observed all the time. For that purpose, the recording software allows repeating of characters, which were not written or segmented correctly.

The segmentation of the recordings was done manually by the subject while recording. The subject had to press a key, before starting and after finishing a character or word. This key press was performed with the other hand than the one used for writing. All subjects were told to have their writing hand in start position for the first stroke of the individual character or word and to hold it still, when pressing the start key. After pressing the start key, the character or word was written in the air. When the end point of the writing motion was reached, the subjects should hold their hand still and press the stop key. This kind of segmentation is not as accurate as using a video stream recorded in parallel but avoids manual postprocessing. The motion between two characters was also recorded for initialization and training of repositioning HMMs used for word modeling. Table 1 shows large differences in recording time, which is probably not only caused by differences in writing speed, but also by differences in segmentation quality.

6. RECOGNITION SYSTEM OVERVIEW

Our system consists of a glove equipped with sensors to measure linear acceleration and angular velocity of hand motion. Sensor data is sent to a computer via Bluetooth. A user can perform 3-dimensional gestures, which are recognized by an HMM classifier based on the delivered signals. Basically any kind of gesture sequence can be modeled. We use character gestures and whole words written in the air to show the potential of our system. Figure 3 shows the basic functional blocks in a diagram.

6.1 Modeling

The HMM modeling was done using the Janus Recognition Toolkit, developed at the Karlsruhe Institute of Technology (KIT) and Carnegie Mellon University in Pittsburgh. As features, we used the accelerometer and gyroscope data, which was normalized by mean and variance resulting in a six-dimensional feature vector per sample. The sampling rate was always set to 819.2 Hz and no filtering was applied to the signals, since the signals are already very smooth. We have made experiments with a moving average filter, which showed no impact. A 0-gram language model was used which allows the recognition of single instances of characters or words out of a defined vocabulary.

6.2 Character Recognition System

For all characters, HMMs with the same topology were used. We always used left-right models with self transitions. Hofmann et al. [6] use HMMs for accelerometer based gesture recognition with a data glove. They evaluate the performance of ergodic and left-right models and can find no significant difference for the task of gesture recognition. The output probability function is modeled by Gaussian Mixtures. The number of states and the number of Gaussians per states was varied on a per experiment basis. That means for one experiment, every character model has the same number of states and Gaussians. Before training, the Gaussians are initialized by k-means clustering of the training data. All training samples are linearly split into as many sample sequences of equal length as the HMM model has states. By that all samples are initially assigned to exactly one HMM state. All samples corresponding to the same state are then collected and clustered in as many clusters as the number of Gaussians in the Gaussian Mixture. Mean and Variance of the clusters are then used as initial values for the GMM. Afterwards the models are trained to maximize their likelihood on the training data using Viterbi training. Different

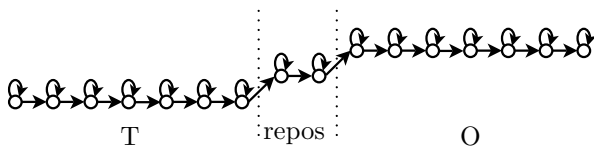


Figure 4: Context model for the word “TO”. It consists of the independent models for the graphemes “T” and “O” with 7 states and the 2 state model for the repositioning in between.

writing variants of one and the same character are modeled by one HMM. There is always one model for one character.

6.3 Word Recognition

Word recognition was performed by concatenating character HMMs to form word models. For a given vocabulary, word models were built from the existing character models. This enables the use of an arbitrary vocabulary. Normally, one has to move the hand from the endpoint of a character to the start point of the next character. To model these repositioning movements, a repositioning HMM is inserted between the character models. We use one left-right HMM to model all repositioning movements. Figure 4 shows an example word model. The initialization and training of these models is the same as for character models. Training data for the repositioning models was collected along with the data for the character models.

6.4 Evaluation Method

On the collected data, two main kinds of experiments have been performed, writer-dependent recognition and writer-independent recognition. For the experiments on writer-dependent recognition, data from one writer was taken and divided into a training, development and test set. On the development set, the parameters, namely number of HMM states and number of Gaussians, were optimized. All writer-dependent results given are from the final evaluation on the test set. Writer-independent recognition was evaluated using a leave-one-out cross validation. We took the data from one writer as test set and the data from the other writers as training data. This is done once for every writer. Again different parameters for the number of states and Gaussians were used and the parameter set was chosen, which gives the best average recognition rate over all writers. No independent test set was used here, due to the small data collection. The performance of all systems is given by the recognition rate, which is defined as the percentage of correctly classified references out of all references. In case of character recognition, a reference is one character, in case of word recognition a reference is one word.

7. EXPERIMENTS AND RESULTS

7.1 Character Recognition

Writer-Dependent. We evaluated writer-dependent recognition performance by testing the recognizer on each writers data individually. For each writer, the data was divided randomly in a training (40%), development (30%) and test (30%) set. The character frequencies were balanced over the sets. A number of 6, 8, 10 and 12 HMM states and 1 to 5

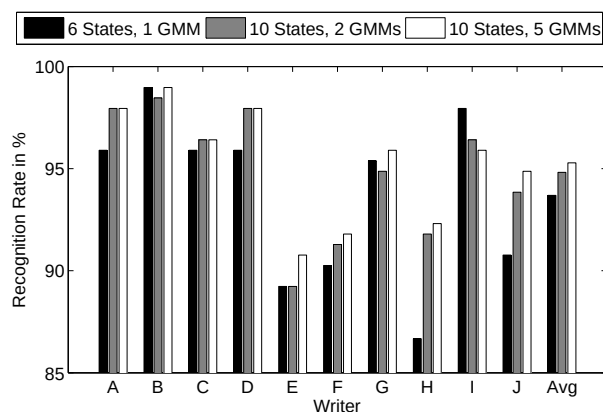


Figure 5: Results of the writer-dependent character recognition for 3 different HMM topologies and number of Gaussians. The worst and best performing systems are shown by black and white bars. To avoid the risk of overspecification, the system represented by the grey bar was used for evaluation.

Gaussians per mixture was investigated. In Figure 5 the results of three exemplary systems with different number of states and Gaussians are shown. The worst performing system with 6 states and 1 Gaussian per state reached an average recognition rate of 93.7% and the best performing system with 10 states and 5 Gaussians per state reached 95.3%. Since the data set is small, overspecification of models is an issue. To avoid the possibility of overspecification, we took a well performing system with few Gaussians in total. Our final evaluation of the writer-dependent system was done with a 10 state model with 2 Gaussians per state. The performance of this system is represented by the grey bar in Figure 5. Table 2 shows the breakdown of results for all writers in the column named WD. An average recognition rate of 94.8% was reached for the writer-dependent case.

Writer-Independent. As written in section 6.4, a leave-one-out cross validation was performed on the collected character data set to investigate the performance of a writer-independent system. The cross validation was done on all data and on the right-hander data only. When models are trained on solely right-hander data and tested on the left-hander data, results are very poor. We repeated all experiments without the left-hander data to exclude this effect. We used HMMs with 8, 10, 12 and 15 states and 2 to 6 Gaussians per state. Parameters were optimized on the data of the left out writer in each cross validation fold. Figure 6 shows the recognition rate dependent on the total number of Gaussians of the system for the right-handers only test. If more than one parameter combination resulted in the same total amount of Gaussians, the range of achieved recognition rates is illustrated by a vertical bar. One can clearly see the performance gain by increasing the number of Gaussians. We used models with 15 states and 6 Gaussians per state for the writer-independent evaluation. Table 2 shows the breakdown of the individual results for the right-hander only evaluation and the evaluation with all ten writers. A recognition rate of 81.9% was reached for the writer-independent

Writer	Recognition Rate		
	WD	WI(RH)	WI(ALL)
A (RH)	97.9	75.4	73.5
B (RH)	98.5	83.5	84.3
C (LH)	96.4		46.8
D (RH)	98.0	90.5	89.8
E (RH)	89.2	86.9	85.4
F (RH)	91.3	77.2	77.1
G (RH)	94.9	80.6	78.8
H (RH)	91.8	72.9	73.8
I (RH)	96.4	86.0	87.8
J (RH)	93.9	84.3	84.6
Average	94.8	81.9	78.2

Table 2: Results of character recognition experiments, writer-dependent and writer-independent. The second column of the table shows the writer-dependent results. The third column shows the writer-independent results when leaving out the left-hander, the fourth column shows the writer-independent results for all writers.

case on the right-hander data.

It is not surprising that the recognition performance drops when testing on the left-handed person, since the writing style of this person differs in a fundamental way from the writing of the right-handed test persons. All horizontal strokes are written in opposite direction and all circular letters are also written in opposite direction.

The main problem of the right-hander only systems are ambiguities in characters and writing variants of different writers. Figure 7 shows the confusion matrix for the cross validation on the right-handed writers. First of all, there are problems with similar graphemes like the pairs (P, D) and (X, Y). The similarities are obvious. In case of P and D , the only difference is the length of the second stroke (the arc). In case of X and Y , depending on how people write it, the only difference is the length of the first stroke (from upper left to lower right). One should notice that the writers did not get any kind of visual feedback on their writing. This probably leads to even more ambiguous characters than when writing with visual feedback. The pair (N, W) is also subject to frequent misclassification. The reason gets obvious when considering the way the character N was written by some test persons. Four of the nine right-handers started the N in the upper left corner. They moved down to the lower left corner and up to the upper left corner again before writing the diagonal stroke. An N written this way has the same stroke sequence than a W . Figure 8 illustrates this ambiguity.

We see that most classification errors arise from the differences in writing style between the individual writers. The test persons do write characters in different ways even under the constraint of block letters. Some of the variants have a very similar motion sequence to variants of different characters observed by other writers. This leads to more ambiguities than in the writer-dependent case. On the single character level, it is hard to solve this problem. But when switching to recognition of whole words, the context

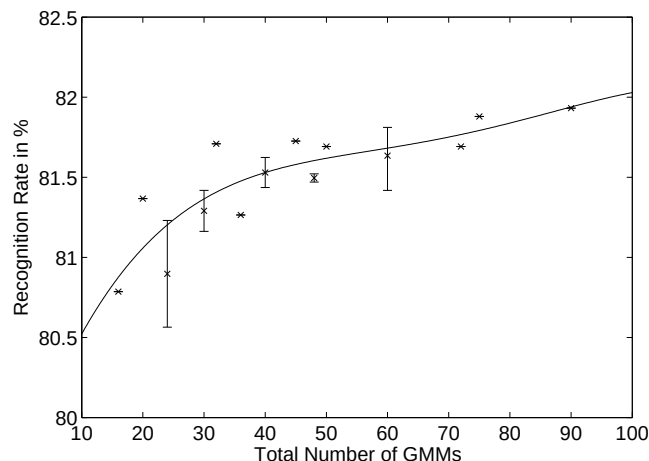


Figure 6: Results of the writer-independent character recognition on right-handers dependent on the total amount of Gaussians per HMM. If different parameter combinations had the same total amount of Gaussians, the performance range is shown as vertical bar. A polynomial fitted on the data illustrates the tendency.

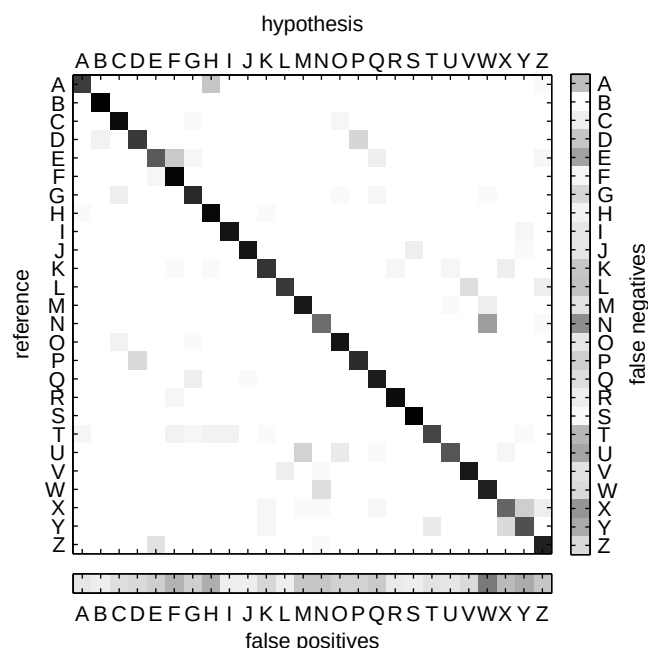


Figure 7: Accumulated confusion matrix for the cross validation of the right-handed writers. The confusion matrices of the tests on each writer were summed together.

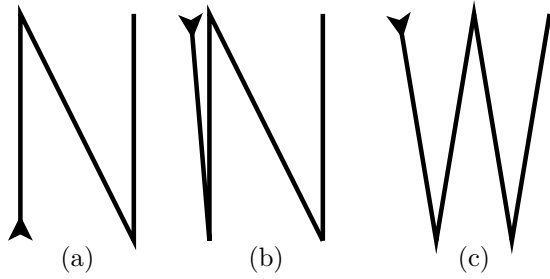


Figure 8: Writing variants of N (a),(b) compared to the stroke sequence of W (c). The allographs (b) and (c) have the same basic stroke sequence.

Data	Features	States	GMM	Rec. Rate
RH	a_{xyz}	15	6	76.8
RH	a_{xyz}, g_{xyz}	15	6	81.9

Table 3: Comparison of sensor configurations on writer-independent character recognition. Accelerometer and gyroscope features are compared to accelerometer only features.

information should help dealing with these ambiguities.

We also investigated the effect of using only accelerometer data as features. We would be able to keep the sensor flat and cheaper if we do not use gyroscopes. We compared the results of using accelerometers (a_{xyz}) and gyroscopes (g_{xyz}) to the results of using only accelerometers. Table 3 shows the results of this comparison. We see, that accelerometer only performance is worse than with the full sensor setup, but depending on the application, this might still be acceptable.

7.2 Word Recognition

The word data was recorded from test person A, who also contributed to the character data set. To build a writer-dependent word recognizer, we took the character data from this test person to initialize the character models. We used the trained models from the experiments on writer-dependent character recognition, i.e. HMMs with 10 states and 2 Gaussians per state. The repositioning models were trained on the repositioning data from the character recording session. We used models with 7 states and 2 Gaussians per state. We randomly split the word data set into two sets (Set A and Set B) of equal size, taking either of the two once as training and once as test set. The vocabulary always contained all 652 words from the recording, that means the recognizer had models for all these words and chose one out of this set as hypothesis. The words were all recorded only once and there were no duplicates in the set. That means, no word in the test set appeared in the training set before. We evaluated the recognizer without any additional training on word recordings and after 10 training iterations on the word training set. Table 4 shows the results of the experiments. Word training boosts performance significantly. The reason is probably two-fold. First, the repositioning models from the character experiments are supposedly not very good, since they were trained with the movements occurring in the artificial pauses between consecutive character recordings. Second, the manual segmentation in the character recording

Train	Test	Word Training Iterations	Recognition Rate
Set A	Set B	0	74.2
Set B	Set A	0	71.8
Average		0	73.0
Set A	Set B	10	97.5
Set B	Set A	10	97.5
Average		10	97.5

Table 4: Results of the writer-dependent word recognition. The character models were already trained on character data. The number of training iterations in the table corresponds only to further training on words.

is inaccurate and by that also character models can profit from word training. We can see that word recognition performance is in the same range than character recognition performance for this writer. Typical misclassifications occur by confusing words that barely differ, like “as” and “was” or “job” and “jobs”.

8. DEMONSTRATOR

Finally we built an online demo system to showcase the applicability of the recognizer. The trained models from the word recognition experiments were taken and a small vocabulary was used. The system can recognize the words necessary to form the sentences “Welcome to the CSL” and “This is our Airwriting System”. The demonstration system uses a laptop with a 2.4 Ghz Intel Core 2 Duo processor and 2 GB RAM for the data recording and the recognition. The system runs stable and few recognition errors occur. A demonstration video of the system can be seen on our website³.

9. CONCLUSION AND FUTURE WORK

We designed and implemented a wearable computer device for gesture based text input. The device consist of a slim data glove with inertial sensors and the ability to transfer sensor data wireless. This enables hands-free operation. We made experiments on character and word recognition. For the task of writer-dependent character recognition, we reached an average recognition result of 94.8%. For the writer-independent case, a recognition rate of 81.9% was reached. We identified ambiguities between characters caused by individual writing style as main reason for the significantly lower recognition rate in the writer-independent case. It will be hard to solve these problems on the single character level. Since our main goal is not single character recognition, but recognition of text, context information should help to solve most of the ambiguities. We implemented a word recognizer using the existing character models from character recognition and reached an average writer-dependent recognition rate of 97.5% for a single test person on a vocabulary of 652 words. We bypass the problems arising from sensor inaccuracies when performing a trajectory reconstruction by working directly on the acceleration and angular velocity values. We show that the introduction of repositioning models between characters is a suitable way to

³<http://csl.ira.uka.de/fileadmin/Demo-Videos/airwriting-chris.mpg>

deal with the lack of pen-up and down information. We implemented a demo system and showed that online operation is feasible.

We plan to extend the word recognition system to be writer-independent by building a word database with different writers. Then we will analyze, if context information really solves most of the ambiguities arising in writer-independent character recognition. We also plan to use a more complex language model, which will enable us to recognize whole sentences.

We will further miniaturize the device by reducing power consumption and with this the size of batteries. We will also further investigate the ability to abandon the gyroscopes.

10. ACKNOWLEDGMENTS

We would like to thank Michael Mende for developing the circuit layout of the electronic components for us and Wolfgang Rihm for the soldering of the components.

11. REFERENCES

- [1] O. Amft, R. Amstutz, Smailagic, A., D. Siewiorek, and G. Tröster. Gesture-controlled user input to complete questionnaires on wrist-worn watches. In *Human-Computer Interaction. Novel Interaction Methods and Techniques*, volume 5611 of *Lecture Notes in Computer Science*, pages 131–140. Springer Berlin / Heidelberg, 2009.
- [2] W.-C. Bang, W. Chang, K.-H. Kang, E.-S. Choi, A. Potanin, and D.-Y. Kim. Self-contained spatial input device for wearable computers. *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.*, pages 26–34, 2003.
- [3] H. Brashear, T. Starner, P. Lukowicz, and H. Junker. Using multiple sensors for mobile sign language recognition. *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.*, pages 45–52, 2003.
- [4] S.-J. Cho and J. Kim. Bayesian network modeling of strokes and their relationships for on-line handwriting recognition. *Pattern Recognition*, 37(2):253–264, 2004.
- [5] A. Hein, A. Hoffmeyer, and T. Kirste. Utilizing an accelerometric bracelet for ubiquitous gesture-based interaction. In *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, volume 5615 of *Lecture Notes in Computer Science*, pages 519–527. Springer Berlin / Heidelberg, 2009.
- [6] F. Hofmann, P. Heyer, and G. Hommel. Velocity profile based recognition of dynamic gestures with discrete hidden markov models. In *Gesture and Sign Language in Human-Computer Interaction*, volume 1371 of *Lecture Notes in Computer Science*, pages 81–95. Springer Berlin / Heidelberg, 1998.
- [7] D. Kim, H. Choi, and J. Kim. 3d space handwriting recognition with ligature model. In *Ubiquitous Computing Systems*, volume 4239/2006 of *Lecture Notes in Computer Science*, pages 41–56. Springer Berlin / Heidelberg, 2006.
- [8] I.-C. Kim and S.-I. Chien. Analysis of 3d hand trajectory gestures using stroke-based composite hidden markov models. *Applied Intelligence*, 15(2):131–143, 2001.
- [9] R.-H. Liang and M. Ouhyoung. A real-time continuous gesture recognition system for sign language. In *Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998. Proceedings.*, pages 558–567, 1998.
- [10] R. McGuire, J. Hernandez-Rebollar, T. Starner, V. Henderson, H. Brashear, and D. Ross. Towards a one-way american sign language translator. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 620–625, 2004.
- [11] J. Oh, S.-J. Cho, W.-C. Bang, W. Chang, E. Choi, J. Yang, J. Cho, and D. Kim. Inertial sensor based recognition of 3-d character gestures with an ensemble classifiers. *Ninth International Workshop on Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004.*, pages 112–117, 2004.
- [12] R. Plamondon and S. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.
- [13] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.
- [14] M. Rehm, N. Bee, and E. André. Wave like an egyptian: accelerometer based gesture recognition for culture specific interactions. In *BCS-HCI '08: Proceedings of the 22nd British HCI Group Annual Conference on HCI 2008*, pages 13–22, Swinton, UK, UK, 2008. British Computer Society.
- [15] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li. Gesture recognition with a 3-d accelerometer. In *Ubiquitous Intelligence and Computing*, volume 5585 of *Lecture Notes in Computer Science*, pages 25–38. Springer Berlin / Heidelberg, 2009.