Optimization of a Handwriting Recognition Algorithm for a Mobile Enterprise Health Information System on the Basis of Real-Life Usability Research

Andreas Holzinger[1,2], Martin Schlögl[1,2], Bernhard Peischl[3], and Matjaz Debevc[4]

[1] Medical University Graz, Institute for Medical Informatics (IMI)
Research Unit HCI4MED, A-8036 Graz, Austria
[2] Graz University of Technology, Institute for Information Systems and Computer Media (IICM), A-8010 Graz, Austria
[3] Graz University of Technology, A-8010 Graz, Institute for Software Technology
[4] University of Maribor, Faculty of Electrical Engineering and Computer Science
SI-2000 Maribor, Slovenia
a.holzinger@hci4all.at, bernhard.peischl@ist.tugraz.at
matjaz.debevc@uni-mb.si

Abstract. Optimizing data acquisition in mobile health care in order to increase accuracy and efficiency can benefit the patient. The software company FERK-Systems has been providing enterprise mobile health care information systems for various medical services in Germany for many years. Consequently, the need for a usable front-end for handwriting recognition, particularly for the use in ambulances was needed. While handwriting recognition has been a classical topic of computer science for many years, numerous problems still need to be solved. In this paper, we report on the study and resulting improvements achieved by the adaptation of an existing handwriting algorithm, based on experiences made during medical rescue missions. By improving accuracy and error correction the performance of an available handwriting recognition algorithm was increased. However, the end user studies showed that the virtual keyboard is still the preferred method compared to handwriting, especially among participants with a computer usage of more than 30 hours a week. This is possibly due to the wide availability of the QUERTY/QUERTZ keyboard.

Keywords: Handwriting recognition, Mobile computer, Human-computer interaction, Usability, Real-life, Health care.

1 Introduction and Motivation for Research

In cases of emergency, rapid patient information collection is very important. This information is most often collected by first aiders (first responders) and paramedics (e.g. Red Cross). Prompt and accurately recorded and well communicated vital patient data can make the difference between life and death [1], [2].

The data acquisition should have as little disruptive effect on the workflow of the emergency responders (rescue staff) as possible. A possible solution for data input can be an mobile application on a lightweight handheld device [3], [4].

M.S. Obaidat, G.A. Tsihrintzis, and J. Filipe (Eds.): ICETE 2010, CCIS 222, pp. 97–111, 2012.

Due to the fact that emergencies are usually within difficult physical situations, special attention to the design of information technology for emergencies has to be taken into consideration [5]. A key issue of any such information system is the acquisition of textual information. However, extensive text entry on mobile devices is principally to be avoided and a simple and easy to use interface, in accordance with the proverb: less is more, is a supreme necessity [6].

The basic evidence that entering data onto a mobile device via a stylus is slower, more erroneous and less satisfactory for end users than entering data via a QWERTZ (de) or QUERTY (us) keyboard has been demonstrated in some studies [7], although, on the other hand the use of a stylus is much faster and more accurate than using finger touch [8]. A specific study for “Ambulance Run Reporting” shows good results for acquiring text with a virtual keyboard, while acquiring text by the application of handwriting recognition showed some serious usability problems [4]. Motivated by this previous work, we focus in this work on handwriting recognition and on how to improve its usability – in case of need, also by adaptation of existing handwriting algorithms. Consequently, in this paper we report on real-life experiences and on some improvements achieved by the adaptation of an existing handwriting engine.

2 Theoretical Background and Related Work

A big difficulty of handwriting recognition is that handwritten characters are variable on an individual basis and that these characters are usually separated into alphabets, numerals, and symbols, despite the different characters of the language itself. Although handwriting recognition will benefit in future from improved adaptive and context-sensitive algorithms, improving the user experience of novice end users with the respective technology is possibly the most important factor in enhancing user acceptance [9]. This is even more important in medical or health care contexts, where the difficulty is in the environmental conditions, e.g. if the person is on the move or in a hurry [10]. Whereas the first problem might be solved by the training modus opportunities, in order to adapt the system to the individual handwriting style, the second problem is only solvable by an extremely robust and usable system. Especially in the health care domain, good end user acceptance and usability can only be obtained by providing simple operation (good user guidance), very short response times and low error rates [11].

Basically, there are several methods for handwriting recognition; these belong basically to two distinct families of classification:

I) Structured and Rule Based Methods

Because of the fuzzy nature of human handwriting, it makes sense to adapt the well known fuzzy logic technique for this purpose [12]. Rather than evaluating the two values as in digital logic, fuzzy terms admit to degrees of membership in multiple sets so that fuzzy rules may have a continuous, rather than stepwise, range of truth of possibility. Therefore non-identical handwritten numerals, from same or different users, can be approximated using fuzzy logic for fast and robust handwriting recognition [13].

II) Statistical Methods

a) Hidden Markov Modeling (HMM)

The attractiveness of HMM for various pattern recognition tasks is mainly due to their clear and reliable statistical framework. Many efficient algorithms for parameter estimation and model evaluation exist, which is an important prerequisite for their practical implementation for real-life applications [14]. The methods using HMM [15], are based on the arcs of skeleton graphs of the words to be recognized and an algorithm applied to the skeleton graph of a word extracts the edges in a particular order, which is transformed into a 10-dimensional feature vector. Each of these features represent information about the location of an edge relative to four reference lines, the curvature and the degree of the nodes incident to the considered edge. Training of the HMM is done by use of the Baum-Welch algorithm, while the Viterbi algorithm is used for recognition [16], [17].

b) Neural Networks

The methods based on Neural Networks were driven by the emergence of portable, pen based computers. A typical approach is to combine an artificial neural network (ANN), as a character classifier, with a context-driven search over segmentation and word recognition hypotheses [18].

However, handwriting recognition not only consists of the recognition itself; the data must undergo some preprocessing:

(I) reduce noise;
(II) normalization, and
(III) segmentation.

The last step, the segmentation phase, segments the input into single characters [19]. Writing discrete characters requires no segmentation; this is done by the users themselves [20].

Another way to improve recognition is to decrease the set of possible alternatives, such as to restrict the set to accepting only lower case letters or digits [21].

system and their current position within its complexity. However, when striving for a design following the "principle of the least surprise", we are faced with the problem that designers and developers rarely are able to predict exactly what the end users really expect (remember Steve Krug [22]: "Don't make me think!").

Efficiency and User Satisfaction were derived. Task Effectiveness (TES) determines how correctly and completely the goals have been achieved in the context.

3 Related Work

To date only a few studies considered handwriting recognition on mobile devices and very few in the health care domain.

A very early work by Citrin et al. report very general on the usage of a pen on a flat surface of a LCD unit (scribing and tapping). They reported that with the maximum rate of 100 selections of direction per second for pen, scribing may produce strokes with the speed of 300 (100×3) bps. However, no more results were found [23].

MacKenzie showed that the recognition accuracy for a set containing upper and lower case letters was lower than for a set containing just lower case letters [24].

Chittaro evaluated a system for recording data on a system during a running ambulance drive, having first responders as participants. Text entry via virtual keyboard and handwriting recognition (MS Transcriber – Calligrapher) were also performed. Text entering by handwriting was considered very laborious and difficult by the users (Mean 3.8, Var 6.6), while entering text by use of the virtual keyboard was quite easy (Mean 7.2, Var 1.8). (0=Hard, 9=Easy). Furthermore, they emphasized the bad usability of entering text by using handwriting recognition. Most words were wrongly recognized and there were enormous problems in correcting those wrongly recognized words [4].

4 Methods and Materials

The aim of our study was to increase the performance of available handwriting recognition by improving accuracy and error correction following solid usability engineering methods [25].

We focused on separate character recognition, since the correction of a single letter, at the moment of false recognition, can be made more naturally, and efficiently, than attempting to correct or delete a single letter within a recognised word.

Due to limited space, there could be some problems inputting long words. Therefore, only one character at a time can be written and recognized.

4.1 Experimental Device

The device used for the prototype was an Asus MyPad A626 PDA (Personal Digital Assistant). This device is equipped with an anti-glare touch screen display. For typing on the touch screen, a stylus is used.

The technical specifications of this device are as follows:

CPU Marvell XScale, 312MHz; Operating System: MS Windows® Mobile™ 6; Memory: 256MB Flash ROM and 64 MB SDRAM; Display: 3.5" Brilliant TFT LCD, 65k full-colours, anti-glare, 16-bit display QVGA, 240 x 320 px touch screen; Weight: 158g; Physical dimensions: 117 mm x 70.8 mm x 15.7cm.

4.2 Dialog Design

The light green area (see figure 1) within the writing sections defines the optimal size for handwritten lowercase characters of 80 points [26].

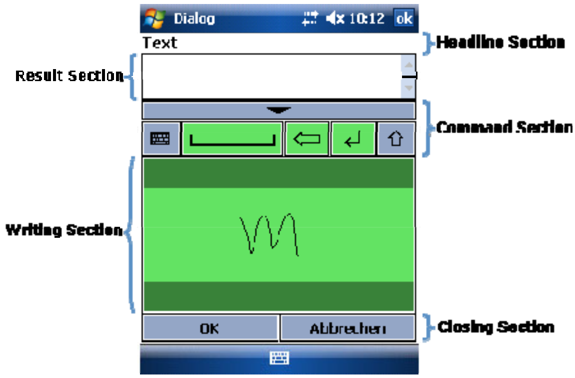

Fig. 1. Design of the handwriting dialog

4.3 Handwriting Recognition

We used the SDK of the handwriting recognition engine Calligrapher (in MS Windows® Mobile Transcriber) in the version 6.0 [26]. This SDK makes it possible to define single character recognition. We can handle the results and a custom timeout (after which time the recognition starts) can be defined.

4.3.1 Adaptive Timeout

A handwritten character consists of one or more strokes. The recognition starts after the character is finished. The system has to await a timeout before starting recognition because the system doesn't know whether the character consists of just one or more strokes. A stroke is defined as the writing from pen down to pen up [20].
Because of the different writing speeds of each user, this timeout has to be calculated for each user. Therefore, the system stores the last ten times which elapse between two strokes.

$$T = \frac{\sum_{i=1}^{11} s(i)}{11} * \frac{X}{100}$$

Fig. 2. Calculation of timeout T [sec]

Figure 2 shows how the timeout is calculated every time a timeout is requested; s(1) is the last calculated average time between strokes, s(2)..s(11) are the last ten stored times between strokes. X is a factor, in this experimental setting X is 200. The result T is the timeout in seconds.

4.3.2 Correction Intervention

Calligrapher SDK 6.0 doesn't adapt recognition on users' handwriting because of the use of static Fuzzy-Neuronal Nets [27].

There are problems with some user's style of writing letters – the user writes a letter (e.g. an "a") but the recognition engine recognizes another letter (e.g. figure 3).

A recognition result is a list of possible characters and its weight (maximum 5 entries). Every time the same letter is wrongly recognized for a user (as in Figure 2), the lists returned by the recognition are similar.

Fig. 3. Written "a" but not recognized as "a", instead as "ir"

These lists (characters and its weight) with its representing letter are stored. Each of them is called schema. During writing, the recognition result will be compared to the stored schemas as follows (see example in Figure 4).

For each stored schema:

Characters from the result list and the list of the schema are compared. If the result list consists of 2 or 3 characters, at least 2 have to match to the stored schemas lists characters. (2 of 2, 2 of 3). If there are 4 or 5 characters in the result list, at least 3 have to match (3 of 4, 3 of 5). This means, the resulting list is validated to the list of the schema. If the list is valid according to the list of the schema, the average deviation between these matching characters is calculated.

Inputted list			Stored Schema's list		Deviation
u	52		k	41	4
n	47		M	35	0
k	37		n	31	16
M	35		h	26	
A	22		m	22	
Validity: 3/5 → VALID					
Average Deviation (16+0+4)/3:					**6**

Fig. 4. Example of a list comparison

The representing letter of the schema with the lowest average deviation will be put in first place of the recognition result.

4.3.3 Calibration

The calibration is designed to collect user specific data for each letter. This data contains weights, which present every character explicitly. Also, schemas of wrongly recognized letters (Chapter 3.3.2) are collected. The system prompts the user to input a letter.

If the result list of the recognition has the prompted letter in first place, the weight will be stored for this letter. In the calibration phase, at least 2 weights will be stored for each letter.

If not, the result list will be stored as a schema with the prompted letter as a representing letter. In the calibration phase, a maximum of 10 schemas for each letter is stored.

This calibration is done once for each user. A continuous calibration is also done during writing in the handwriting recognition dialog, saving weights and schemas for correctly recognized letters (but not for deleted letters).

4.3.4 Other Interventions on Recognition Results

To avoid side effects, the intervention described in Chapter 3.3.2 is only made when the weight of a recognized letter is less than the average weight for this letter (average of the weights for this letter collected by calibration).

Other interventions are made to avoid potential. problems with highly confusable pairs such as "r" and "v" [21]. (I) While writing a word, only letters and punctuation marks are valid, recognized results. (II) Just deleted letters (with BACKSPACE) are not valid, recognized results for the next recognition (III) Special handling for "O" and "0" as first letter of a word or number.

4.4 Experiment

The real life environment is mostly a seat in an ambulance car (refer to figure 5). To avoid negative effects on ambulance responder's work, the experiment is done in their recess in the ambulance service rooms, simulating the circumstances (sitting in a car) by doing the experiment sitting on a chair, holding the PDA in their hand, without laying down the elbows on e.g. an armrest. [28] shows that simulating environments gives almost the same results.

Fig. 5. Participants from the ambulance service during experiments in real life

Participants were people who work as ambulance officers (professionals, volunteers and former civilian service). No previous experience with mobile computers was required. They were asked to fill out a background questionnaire to obtain data about their age, education and use of computers. The prototype for the experiment is divided into two parts, one for virtual keyboard based text input, and the other for handwriting recognition input. Within these two parts, the users have the opportunity to become familiar with the input methods. After that, the user has to input a given text to the experimental dialog (for measuring the accuracy). Due to measuring the accuracy, text entry is done as text copy [29]. This text consists of 13 German words (94 characters without spaces, 106 with spaces). After the keyboard based experimental dialog, the calibration of the handwriting is done. Speed in wpm, words per minute [24, 30] and the accuracy of the handwriting recognition are measured and calculated. At the end, a feedback questionnaire is filled out by the user. Some questions are based on the study of Chittaro [4]. Every single test user conducted the procedure outlined in Figure 6.

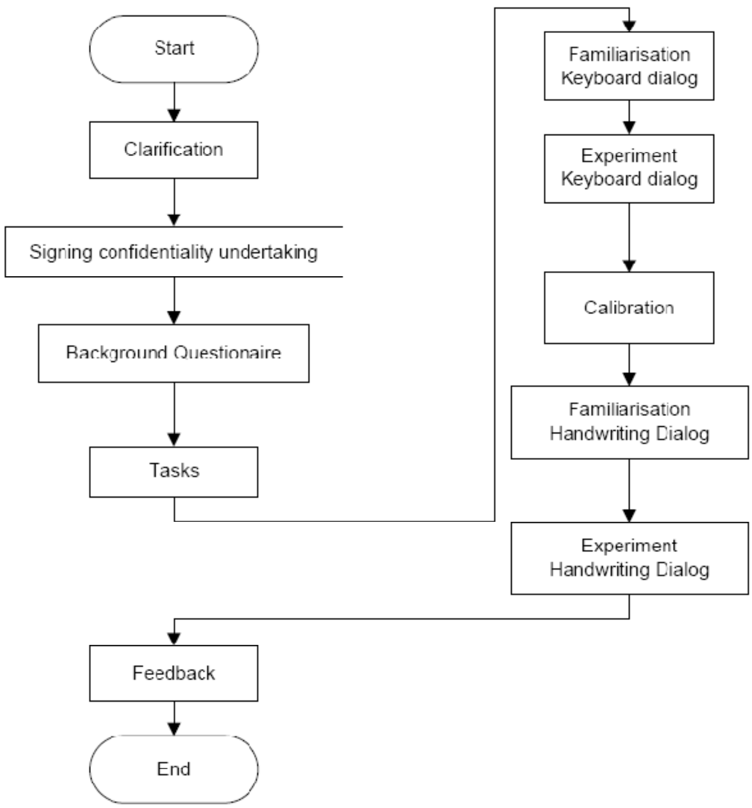

Fig. 6. The procedure for the experiment

5 Results

The participants of the experiment were professional (9) and volunteer (8) first responders of the Austrian Red Cross, one student of medicine and three others (because everyone could be a volunteer first responder, not only medical students). 10 were experienced on a PDA or a mobile phone with touch screen, while 11 had no experience with touch screens.

Their ages ranged from 20 to 85 years. Two elderly people (68 and 85 years) were chosen. The reason for this was, that we wanted to gain insight into the behaviour and performance of somebody who had never before used a QWERTY/QUERTZ keyboard or a PC in their life before. The average use of a PC was 12.3 years, using a PC 31 hours in average per week. 11 participants used a PC $\leq$ 30 hours a week, while 10 participants used a PC for more than 30 hours. One of the 21 participants was left-handed. All participants had normal or corrected to normal eyesight and none suffered of any kind of colour blindness.

5.1 Accuracy

Overall		≤ 30 weekly usage		> 30 weekly usage	
Mean	Var	Mean	Var	Mean	Var
99.1	6.28	100	11.5	99.06	1.44

Fig. 7. Table: Accuracy inputting text with virtual keyboard [%]; all participants, participants ≤ 30 hours and above

Overall		≤ 30 weekly usage		> 30 weekly usage	
Mean	Var	Mean	Var	Mean	Var
89.25	34.3	91.43	30.20	88.00	37.34

Fig. 8. Table: Recognition accuracy [%] of handwriting recognition; all participants, participants ≤ 30 hours and above with interventions

Overall		≤ 30 weekly usage		> 30 weekly usage	
Mean	Var	Mean	Var	Mean	Var
84.66	57.6	86.99	79.15	83.33	38.21

Fig. 9. Table: Recognition accuracy [%] of handwriting recognition; all participants, participants ≤ 30 hours and above without interventions

The participants using a PC ≤ 30 hours a week include the two elderly people.

The 85 year old participant has an accuracy of 89.2% for inputting text with the virtual keyboard and a recognition accuracy of 80.1% with interventions and 65.6% without interventions.

The 68 year old participant had an accuracy of 100% for inputting text with the virtual keyboard and a recognition accuracy of 95% with interventions and 90.8% without interventions.

The 85 years old participant has an accuracy of 89.2% for inputting text with the virtual keyboard and a recognition accuracy of 80.1% with interventions and 65.6% without interventions.

The 68 year old participant has an accuracy of 100% for inputting text with the virtual keyboard and a recognition accuracy of 95% with interventions and 90.8% without interventions.

5.2 Speed

Participants using a PC ≤ 30 hours a week include two elderly people. The 85 year old participant wrote 2.87 wpm with the keyboard and 2.82 wpm with handwriting recognition. The 68 year old participant wrote 4.88 wpm with the keyboard and 4.17 wpm with handwriting recognition.

Overall		≤ 30 weekly usage		> 30 weekly usage	
Mean	Var	Mean	Var	Mean	Var
13.17	27.7	12.88	29.46	13.43	18.29

Fig. 10. Table: Words per minute virtual keyboard; all participants, participants ≤ 30 hours and above

Overall		≤ 30 weekly usage		> 30 weekly usage	
Mean	Var	Mean	Var	Mean	Var
8.44	4.59	8.11	5.37	8.71	1.95

Fig. 11. Table: Words per minute handwriting recognition; all participants, participants ≤ 30 hours and above

5.3 User Questionnaire

Overall	**Mean**	**Var**
Keyboard		
Inputting Data (+4=easy, -4=difficult)	3.0	2.6
Correction of wrong inputted data (+4=easy, -4=difficult)	4.0	2.8
Handwriting		
Inputting Data (+4=easy, -4=difficult)	2.0	4.9
Correction of wrongly input/recognized data (+4=easy, -4=difficult)	3.0	1.4
Did the recognition slow down your writing (+4=no, -4=yes)	0.5	9.2
I would prefer (+4=handwriting, -4=keyboard)	-2.5	7.9
Basic Information		
Use of colour is (+4=useful, -4=useless)	2.0	2.5
The handwriting recognition positively surprised me (+4=yes, -4=no)	2.5	7.1
Characters on the PDA are easy to read (+4=yes, 4=no)	4.0	3.9

Fig. 12. Overall results of user questionnaire

Weekly computer usage [hours]	**Mean (<=30)**	**Var (<=30)**	**Mean (>30)**	**Var (>30)**
Keyboard				
Inputting Data	3.5	1.4	3.0	3.7
Correction of wrongly input data	4.0	1.7	3.5	4.5
Handwriting				
Inputting Data	2.5	4.2	0.5	3.8
Correction of wrongly input/recognized data	4.0	0.5	2.5	1.8

Fig. 13. Results of user questionnaire for weekly usage of computer ≤ 30 hours and above

Did the recognition slow down your writing	2.5	8.4	-0.5	8.2
I would prefer	-1.5	2.0	-2.5	4.9
Basic Information				
Use of colour	2.0	2.5	1.5	2.6
The handwriting recognition positively surprised me	4.0	7.7	0.0	4.7
Characters on the PDA are easy to read	4.0	4.6	4.0	3.2

Fig. 13. (*Continued*)

5.4 Timeout

Figure 14 outlines the timeout for each user during the experiment described previously. The figure lists the optimized timeout for each of our test users. Most notably, the optimal timeout for the majority of our test users is between 200 and 250 milliseconds. The algorithm is capable of adapting the timeout to slow as well as to fast writers (e.g. user 19 is a rather slow writer whereas user 5 is the fastest one).

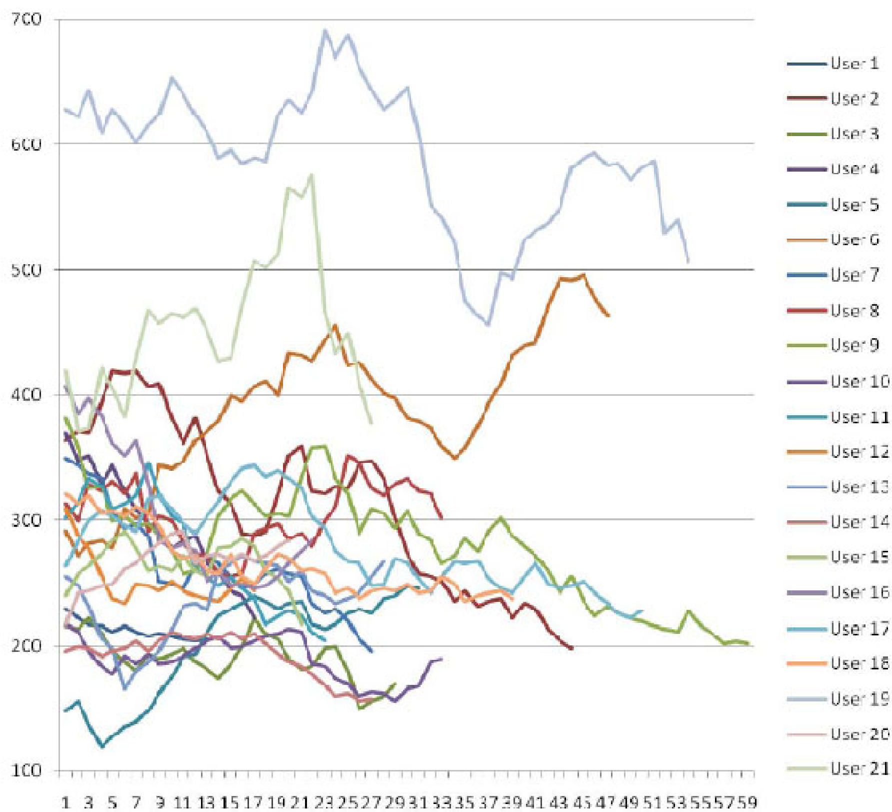

Fig. 14. Timeout adaption for each user

6 Discussion and Lessons Learned

Entering text with the virtual keyboard (Mean 3.0, Var 2.6) was easier for the participants than with handwriting (Mean 2.0, Var 4.9). However, compared to the study of [4], we could reach an significant improvement by inputting data with handwriting. Interestingly, inputting data by handwriting recognition was rated easier by participants who use computers less than or equal to 30 hours a week than by participants with extensively more use (Mean 2.5; Var 4.2; against Mean 0.5, Var 3.8 of virtual keyboard). Also, the correction on the handwriting recognition dialog was rated easier (Mean 4.0, Var 0.5; against Mean 2.5, Var 1.8; of virtual keyboard). Participants with a computer usage of more than 30 hours a week preferred the virtual keyboard (Mean -2.5, Var 4.9) more than the other participants (Mean -1.5, Var 2.0). This could be a result of hardly any handwriting during work and much more typing text on classical keyboards (QWERTZ or QUERTY). Consequently, the two elderly participants were included in this study, in order to obtain data regarding participants who never used any computer or handheld device. The elderly participants were the only ones who provided a complete preference to the handwriting recognition in contrast to the virtual keyboard. This is also clearly visible in the results for these participants, although both groups have quite comparable results in wpm for the virtual keyboard and the handwriting text input.

This is an interesting result; however, it is not of practical relevance, since there are hardly any people left – at least amongst people able to volunteer as a first responder – without experience on computer keyboards. Today, from elementary school on, children get used to work with computers by using the QWERTZ or QUERTY keyboard.

Nevertheless, our interventions on the basis of the results of the handwriting recognition, finally paid off in an significant improvement on the recognition accuracy (over all participants a better accuracy of Mean +4.39%, Var 9.54).

These interventions can also be useful for the improvement of other handwriting recognition engines, due to the fact that our interventions were only made on the results of the engine, achieving better accuracy. The use of a handwriting recognition engine with a higher accuracy than e.g. Calligrapher, in combination with our demonstrated interventions, may even improve the overall accuracy. Our methods on operating on the results of the handwriting recognition engine operate context independent. Using a dictionary to add the likelihood of upcoming characters may improve the accuracy in that part of the problem regarding confusable pairs, such as “r” and “v”. Because of typing in characters one by one, a word completion feature could be added to handwriting recognition too. This also would increase the writing speed.

7 Conclusion and Future Outlook

Although much research in the field of handwriting recognition has been done, recognition algorithms still do not 100 % achieve the high prospects of the users. Handwriting is very individual to every person and identifying characters is still very hard – as long time ago described by [31]. This paper demonstrated operations on the result of a recognition engine. Replacing the engine used in this experiment (Calligrapher) with a better recognition engine with higher accuracy can improve the result on accuracy due to the fact that accuracy is the most important parameter for the acceptance. Users rate an accuracy of 90% to 95% to be a very poor recognition rate. Users will accept only accuracies more than 97% (3 wrong recognized items out of 100 inputs). LaLomia [32], [33] let users write specified text to a system, which immediately interpreted the input with randomly occurring errors. The randomly generated accuracy ranged from 90% to 99%. After reviewing the errors, the users rated the acceptability; exclusively recognition accuracy over 97% was acceptable. Because of typing in characters one by one, a word completion feature could be added to the final implementation, however, in the last years, mobile devices and the usage of such devices changed rapidly. Nowadays, many people, especially younger people are connected to social networks, including Facebook, especially by using their smart phones – where today the user interface consists of a touch screen. Data acquisition is mostly realized with improved, intelligent virtual keyboards. Such intelligent keyboards e.g. with implementation of a regional error correction [34]. Often they are connected with tactile feedback for touch screen widgets [35] which can improve performance and usability of virtual keyboards on small screens. Handwriting is taught from elementary school on and nearly everyone learns handwriting at school. Therefore, handwriting recognition is a very important technology for input interfaces of mobile computers. However, today, even children get used to the QWERTY layout keyboard from elementary school. Consequently, interface designers can assume that nearly everyone is experienced in using a QWERTY layout keyboard.

Because of the higher user acceptance of current uncomfortable virtual keyboards compared to handwriting recognition, future developments and projects should focus on data acquisition based on intelligent, comfortable virtual keyboards.

Acknowledgements. This study was performed with support of FERK-Systems. We cordially thank the engineering team for their continued and effective industrial support of this work. The research was partially funded by the Austrian Research Promotion Agency (FFG) within one „Innovationsscheck Österreich".

References

1. Holzman, T.G.: Computer-human interface solutions for emergency medical care. Interactions 6(3), 13–24 (1999)
2. Anantharaman, V., Han, L.S.: Hospital and emergency ambulance link: using IT to enhance emergency pre-hospital care. International Journal of Medical Informatics 61(2-3), 147–161 (2001)
3. Baumgart, D.C.: Personal digital assistants in health care: experienced clinicians in the palm of your hand? The Lancet 366(9492), 1210–1222 (2005)
4. Chittaro, L., Zuliani, F., Carchietti, E.: Mobile Devices in Emergency Medical Services: User Evaluation of a PDA-Based Interface for Ambulance Run Reporting. In: Löffler, J., Klann, M. (eds.) Mobile Response 2007. LNCS, vol. 4458, pp. 19–28. Springer, Heidelberg (2007)
5. Klann, M., Malizia, A., Chittaro, L., Cuevas, I. A., Levialdi, S.: HCI for emergencies. In: CHI 2008 Extended Abstracts on Human Factors in Computing Systems, pp. 3945–3948 (2008)

6. Holzinger, A., Errath, M.: Mobile computer Web-application design in medicine: some research based guidelines. Universal Access in the Information Society International Journal 6(1), 31–41 (2007)
7. Haller, G., Haller, D.M., Courvoisier, D.S., Lovis, C.: Handheld vs. Laptop Computers for Electronic Data Collection in Clinical Research: A Crossover Randomized Trial. Journal of the American Medical Informatics Association 16(5), 651–659 (2009)
8. Holzinger, A., Höller, M., Schedlbauer, M., Urlesberger, B.: An Investigation of Finger versus Stylus Input in Medical Scenarios. In: Luzar-Stiffler, V., Dobric, V.H., Bekic, Z. (eds.) ITI 2008: 30th International Conference on Information Technology Interfaces, pp. 433–438. IEEE (2008)
9. MacKenzie, I.S., Chang, L.: A performance comparison of two handwriting recognizers. Interacting with Computers 11(3), 283–297 (1999)
10. Holzinger, A., Hoeller, M., Bloice, M., Urlesberger, B.: Typical Problems with developing mobile applications for health care: Some lessons learned from developing user-centered mobile applications in a hospital environment. In: Filipe, J., Marca, D.A., Shishkov, B., Sinderen, M. v. (eds.) International Conference on E-Business (ICE-B 2008), pp. 235–240. IEEE (2008)
11. Holzinger, A., Geierhofer, R., Searle, G.: Biometrical Signatures in Practice: A challenge for improving Human-Computer Interaction in Clinical Workflows. In: Heinecke, A.M., Paul, H. (eds.) Mensch & Computer: Mensch und Computer im Strukturwandel, Oldenbourg, pp. 339–347 (2006)
12. Gader, P.D., Keller, J.M., Krishnapuram, R., Chiang, J.H., Mohamed, M.A.: Neural and fuzzy methods in handwriting recognition. Computer 30(2), 79–86 (1997)
13. Shi, B., Li, G.: VLSI Neural Fuzzy Classifier for Handwriting recognition (2006)
14. Plotz, T., Fink, G.A.: Markov models for offline handwriting recognition: a survey. International Journal on Document Analysis and Recognition 12(4), 269–298 (2009)
15. Marti, U.V., Bunke, H.: Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition systems. In: Hidden Markov Models: Applications in Computer Vision, pp. 65–90. World Scientific Publishing Co., Inc. (2002)
16. Bunke, H., Roth, M., Schukattalamazzini, E.G.: Off-Line Cursive Handwriting Recognition Using Hidden Markov-Models. Pattern Recognition 28(9), 1399–1413 (1995)
17. Xue, H.H., Govindaraju, V.: Hidden Markov models combining discrete symbols and continuous attributes in handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(3), 458–462 (2006)
18. Yaeger, L.S., Webb, B.J., Lyon, R.F.: Combining Neural Networks and Context-Driven Search for On-Line, Printed Handwriting Recognition in the Newton. In: Orr, G.B., Müller, K.-R. (eds.) NIPS-WS 1996. LNCS, vol. 1524, pp. 275–298. Springer, Heidelberg (1998)
19. Plamondon, R., Srihari, S.N.: On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 63–84 (2000)
20. Tappert, C.C., Suen, C.Y., Wakahara, T.: The State of the Art in On-Line Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(8), 787–808 (1990)
21. Frankish, C., Hull, R., Morgan, P.: Recognition accuracy and user acceptance of pen interfaces. In: Conference on Human Factors in Computing Systems, pp. 503–510 (1995)
22. Krug, S.: Don't Make Me Think: A Common Sense Approach to Web Usability. New Riders, Indianapolis (2000)

23. Citrin, W., Halbert, D., Hewitt, C., Meyrowitz, N., Shneiderman, B.: Potentials and limitations of pen-based computers. In: Proceedings of the 1993 ACM Conference on Computer Science, pp. 536–539 (1993)
24. MacKenzie, I.S., Nonneke, B., Riddersma, S., McQueen, C., Meltz, M.: Alphanumeric entry on pen-based computers. International Journal of Human-Computer Studies 41(5) (1994)
25. Holzinger, A.: Usability Engineering for Software Developers. Communications of the ACM 48(1), 71–74 (2005)
26. Phatware: Calligrapher SDK 6.0 Developer's Manual (2002)
27. Strenge, M.: Konzepte und Toolkits zur Handschrifterkennung (2005)
28. Kjeldskov, J., Skov, M.B., Als, B.S., Høegh, R.T.: Is It Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in Field. In: Brewster, S., Dunlop, M.D. (eds.) Mobile HCI 2004. LNCS, vol. 3160, pp. 61–73. Springer, Heidelberg (2004)
29. MacKenzie, I.S., Soukoreff, R.W.: Text Entry for Mobile Computing: Models and Methods, Theory and Practice. Human-Computer-Interaction 17(2), 147–198 (2002)
30. Lewis, J.R.: Input Rates and User Preference for three small-screen input methods: Standard Keyboard, Predictive Keyboard and Handwriting Human Factors and Ergonomics Society (1999)
31. Neisser, U., Weene, P.: A note on human recognition of hand-printed characters. Information and Control 3, 191–196 (1960)
32. LaLomia, M.J.: User acceptance of computer applications with speech, handwriting and keyboard input devices. Posters and short talks of the, SIGCHI Conference on Human Factors in Computing Systems, pp. 58–58 (1992)
33. LaLomia, M.: User acceptance of handwritten recognition accuracy. In: Conference Companion on Human Factors in Computing Systems, pp. 107–108 (1994)
34. Kwon, S., Lee, D., Chung, M.K.: Effect of key size and activation area on the performance of a regional error correction method in a touch-screen QWERTY keyboard. International Journal of Industrial Ergonomics 39(5), 888–893 (2009)
35. Koskinen, E., Kaaresoja, T., Laitinen, P.: Feel-good touch: finding the most pleasant tactile feedback for a mobile touch screen button. In: Proceedings of the 10th international conference on Multimodal interfaces, pp. 297–304 (2008)