HANDWRITING RECOGNITION ON MOBILE DEVICES
State of the Art Technology, Usability and Business Analysis

Andreas Holzinger, Lamija Basic
Research Unit Human-Computer Interaction, Institute for Medical Informatics, Medical University of Graz, Graz, Austria

Bernhard Peischl
Softnet, Graz, Austria

Matjaz Debevc
Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Slovenia

Keywords: Handwriting recognition, Mobile computer, Human-computer interaction, Usability engineering, Healthcare.

Abstract: The software company FERK-Systems has been providing mobile health care information systems for various German medical services (e.g. Red Cross) for many years. Since handwriting is an issue in the medical and health care domain, a system for handwriting recognition on mobile devices has been developed within the last few years. While we have been continually improving the degree of recognition within the system, there are still changes necessary to ensure the reliability that is imperative in this critical domain. In this paper, we present the major improvements made since our presentation at the ICE-B 2010, along with a recent real-life usability evaluation. Moreover, we discuss some of the advantages and disadvantages of current systems, along with some business aspects of the vast, and growing, mobile handwriting recognition market.

1 INTRODUCTION AND MOTIVATION FOR RESEARCH

In the first quarter of 2010, sales of mobile devices grew by 56.7% according to figures from the International Data Corporation (IDC); the premier provider of market intelligence. These numbers are outpacing the 21.7% growth of the overall mobile market. The majority of smartphones are tailored toward the business-to-consumer (B2C) market, thus the predominant input technique for mobile devices is the multi-touch concept (Wang and Ren, 2009).

Despite these facts from the consumer market, medical professionals (medical doctors, nurses, therapists, first responders etc.) are more familiar with dictation and handling a stylus, since they are used to handling a pen all the time (Holzinger et al., 2008b), (Holzinger et al., 2008a).

As regards the input technology, the most recent development on the mobile market is in contrast to the preferred input technique of professionals in the medical domain. Whereas, from the view-point of Human-Computer Interaction (HCI), handwriting can be seen as a very natural input technology (Holzinger et al., 2006), studies have shown that a recognition rate below 97% is not acceptable to end users (Lee, 1999). The challenge in developing such a system is the fact that the art of handwriting is very individual for everybody, making a universal recognition of all handwriting particularly demanding. In this paper, we extend our experiences from ICE-B 2010 (Holzinger et al., 2010) and present our improvements of handwriting recognition on mobile devices. Moreover, we discuss business issues of current handwriting recognition systems on mobile devices.

Holzinger A., Basic L., Peischl B. and Debevc M..
HANDWRITING RECOGNITION ON MOBILE DEVICES - State of the Art Technology, Usability and Business Analysis.
DOI: 10.5220/0003522102190227
In *Proceedings of the International Conference on e-Business* (ICE-B-2011), pages 219-227
ISBN: 978-989-8425-70-6

2 THEORETICAL BACKGROUND

Handwriting recognition is still considered as an open research problem mainly due to its substantial individual variation in appearance, consequently the challenges include the distortion of handwritten characters, since different people may use different style of handwriting, direction etc. (Perwej and Chaturvedi, 2011).

If a system needs to deal with the input of different end users, a training phase is required to enable the system to understand the user's art of writing. The data received in this phase is stored in a database. During the recognition process, the system compares the input with the stored data and calculates the output.

Basically, handwriting recognition can be separated into online and offline recognition.

I) Offline Handwriting Recognition

Offline recognizers have not received the same attention as online recognizers (Plotz and Fink, 2009).

There are several problem areas (e.g. postal address recognition) where offline handwriting recognizers are very useful due to the large amount of hand written text.

These systems have the ability to convert text into image form. The main disadvantage is that there is no possibility of obtaining information about the type of the input.

First, the text has to be separated into characters or words. With Hidden Markov Models or Neural Networks these words are matched to a sequence of data (Graves and Schmidhuber, 2009). Most recently a work based on hybrid statistical features has been published (Sulong, Rehman and Saba, 2010).

II) Online Handwriting Recognition

These systems collect data during the process of input. The advantage is that specific information, such as the number of used strokes, can be collected. The result is calculated in real time (Liu, Cai and Buse, 2003).

This kind of recognition is mostly used in communication devices such as Smartphones or PDAs. In this paper, we concentrate on the online handwriting recognition technique (Dzulkifli, Muhammad and Razib, 2006) and present a detailed review of techniques and applications for online cursive handwriting recognition.

The first part of this article deals with the review of the main approaches employed in character recognition, since most of these are also used in cursive character recognition.

III) Recognition Process

Most recognition systems comprise of four distinct recognition phases (Liu, Cai and Buse, 2003):

(1) Preprocessing: In this step, noise and other undesirable effects are reduced to improve the data for the recognition process (Liu, Cai and Buse, 2003). Typically, some form of noise reduction and size normalization is applied.

Noise Reduction: During the input, undesired data can also be registered. For example, if the user accidently touches the screen. Such "wild points" have to be corrected.

Size Normalization: During the input the size of a character can vary. For a better recognition the characters have to be normalized to a general size.

(2) Feature Extraction: In this step, the relevant information from the input is extracted. The challenge is to extract a minimal set with maximum data recognition.

(3) Classification and (4) Recognition: The goal is to find the optimal letter to a given sequence of observations. The letter corresponding to the maximum probability is reported as the recognized letter (Plamondon and Srihari, 2000), (Shu, 1997). Compared with other techniques, Neural Networks and Hidden Markov Models are more often used for handwriting recognition (Zafar, Mohamad and Othman, 2005).

Basically, we distinguish between statistical methods (relying on Hidden Markov models or neural networks) and structured and rule-based methods including the following:

Statistical Methods

Hidden Markov Model: HMMs consist of two processes. The underlying process is hidden and contains the state. The observable process contains the output which is visible.

The states have probability distributions over the possible output tokens. The further behavior of the system depends on its present state (Plamondon and Srihari, 2000).

HMMs based on word models have the problem that the model set can grow quite large. Because of this, systems using letter models have become very popular.

Neural Networks (NNs): This method for classification has become popular since the 1980s (Graves and Schmidhuber, 2009). NNs consist of multiple layers (input, output and hidden). Feed-forward neural networks are mostly used. The ability to train an NN and the back propagation of errors are the main advantages. A comparative study regarding

NNs for online handwritten character recognition was conducted by (Zafar, Mohamad and Othman, 2006).

Fuzzy Logic (FL): Each Fuzzy system is realized in three steps.
1) Fuzzification: Based on the features extracted in the further step the fuzzy sets could be generated easily.
2) Rule Application: The fuzzy sets are evaluated with the rules written for the system.
3) Defuzzification: In the last phase the output is generated (Gowan, 2004), (Gader et al., 1997).

3 RELATED WORK

In the following, we briefly discuss the work in relation to the most notable products in handwriting recognition and list the major advantages and drawbacks.

Calligrapher SDK: The application, which we have developed and present in this paper is based on the use of Calligrapher SDK (Phatware, 2008).
This recognition technology uses fuzzy logic and neuronal networks. Calligrapher is based on an integrated dictionary, which is used for the modeling process. It recognizes dictionary words from its main user-defined dictionary, as well as non-dictionary words, such as names, numbers and mixed alphanumeric combinations. The Calligrapher SDK provides automatic segmentation of handwritten text into words and automatically differentiates between vocabulary and non-vocabulary words, and between words and arbitrary alphanumeric strings. Further it supports several styles of handwriting, such as cursive, print and a mixed cursive/print style.

Advantages: The application provides many possibilities.
Disadvantages: The main problem is that it cannot be adapted to a specific end user.

Microsoft Tablet PC: This recognizer works with the Optical Character Recognition and the Convolutional Neural Networks. Such Neural Networks do not need feature vectors as input. The Tablet PC is also able to adapt to a new user during a training phase (Pittman, 2007).

Advantages: The system provides many possibilities. There is a higher recognition rate of subsequently entered words because the detection depends on an integrated Dictionary.
Disadvantages: Users are given many unsolicited hints in order to use the device properly. This suggests that the adjustment to the user is not working very well and disrupts smooth functioning.

WritePad: This is a handwriting recognition system developed for iPhone, iPod and iPad Touch devices. The user can write directly onto the display using a finger or an AluPen. WritePad can recognize all styles of writing. It adapts to the user's style of writing, so it takes time until the user can use it with a lower error rate. Furthermore, it has an integrated shorthand feature, which allows the user to enter frequently used text quickly. To use the system properly, Apple offers an exhaustive tutorial. The user has to write large and clearly for a correct translation. WritePad also includes an auto-corrector, however, this currently supports only English (Phatware, 2008).

Advantages: Through the training phase, the system can adapt to the writing style of the user.
Disadvantages: The user needs patience because the learning process can take longer in some circumstances.

HWPen: HWPen is a handwriting recognition tool which has already been published in 2008 for Apple devices. The software was developed by the company Hanwang.com.cn, mainly for the Chinese language. The system is heavily based on Graffiti. The adjustment period is longer because the user has first to learn the art of writing. However, the system works, similar to Graffiti, very efficiently later (Bailey 2008) (HWPen 2008).

Advantages: Since all characters differ greatly, HWPen has a very good detection.
Disadvantages: The user has to learn a new way of writing.

CellWriter: This is an open source HWR-System for Linux. CellWriter is based on the user's style of writing.

Therefore, a training session must be completed before use. Each character must be written in a separate cell. The system provides a drop-down list of other matches if the recognized result is wrong (Willis, 2007).

Advantages: It provides a word recognition feature.
Disadvantages: CellWriter is only available for Linux.

MyScriptStylus: This HWR-System is based on the latest version of MyScript and can run on Windows, Mac and Linux. The software can recognize about 26 different languages. It provides a lot of different modes, such as Writing Pad mode, in which all kinds of writing (cursive, digit, hand printed) can be recognized. For a better recognition the Character Pad mode can be used, which works similar as CellWriter, whereby the user has to input the letters in cells. Even if the system can work without a training phase, a personal dictionary should be created for better accuracy. This software also provides a list of alternatives in the case of a wrong recognition (VisionObjects, 2009).

Advantages: A lot of language packages and different styles are provided.
Disadvantages: The activation code for the use costs about 40€ (without the calculator module).

Except for Graffiti and HWPen, all of the described systems try to give the user as much freedom in writing as possible. However, this leads to an accuracy rating worse than that of strict systems.

On the other hand, the big disadvantage of recognition systems like Graffiti is that the user has to learn a totally new art of writing.

No matter which path one follows, in both cases the user has to work with the device for some time to learn how to write clearly and precisely. This is the reason why HWR-Systems are not widely accepted as the majority of the users typically do not want to spend much time for the learning phase.

4 METHODS AND MATERIALS

The aim of this work was to continuously develop the system based on our previous development of an emergency medical notation system (Holzinger et al., 2010). The developed system works with character recognition and uses Calligrapher SDK, version 6.0, as the recognition engine. In addition to the recognition of Calligrapher, a novel intervention mechanism was developed to improve the result of an input.

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. Table 1 contains the technical specifications.

Table 1: Specifications of the PDA ASUS MyPal A626.

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, 240x320 px touch screen
Weight	158g
Physical Dimensions	117 mm x 70.8 mm x 15.7cm

4.2 Dialog Design

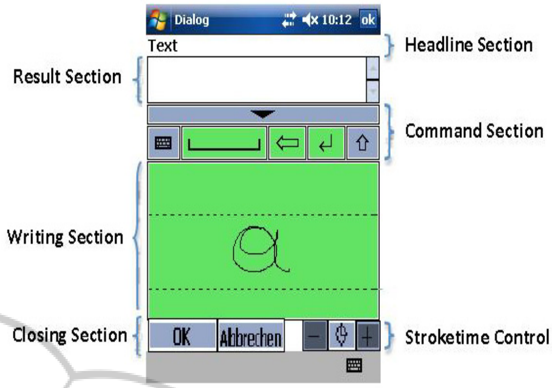

Figure 1: The improved design of the handwriting dialog, compare with the previous design published in (Holzinger et al., 2010).

The lower case letters must be viewed as three separate groups. The so-called “high” letters, such as f, h, b, …; the so-called “low” or tailed letters: p, g, q, …; and the “middle” letters, with neither tails nor uprights: a, c, e, r, etc. This third group is meant to be written in the middle of the green box shown above. Letters from the first two groups, exceeding the given space, are supposed to go above, respectively below, in the remaining highlighted green area of the diagram (e.g. b, q).

4.3 Adaptive Timeout

During the entry of a letter a pre-calculated waiting period (pause) should occur. This prevents the Calligrapher from translation until the user has finished his entry. A character can consist of one or more strokes. Considering this fact the pause must not only relate to the deduction of the pen. Due to the fact that each user enters characters with a different speed the pause should be calculated individually.

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

Figure 2: The calculation pause T [sec] (Holzinger et al., 2010).

In Figure 2, we see how the pause is calculated; every time a pause 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 pause in seconds (Holzinger et al., 2010).

The longer a user works with the unit the faster he will become. Therefore, it is possible that the timeout can be changed manually too.

This feature is not available in the old version. After every ten strokes the timeout was recalculated. However, the main disadvantage thereof is that a rapid change in velocity is not immediately accepted.

Example: A user writes very fast. The pause is therefore rather short. If he begins to write more slowly, it could be that the translation starts too fast. We would have to wait for the recalculation of the pause.

4.4 Calibration

The Calibration is the core of the application. All necessary data is collected and stored here.

In the old version, all letters were typed twice. If the letter was not clearly detected, it was attached to the Calibration again. This could lead to as many as 10 schemata for a letter. Furthermore, the calibration was continued during the writing in the handwriting calibration dialog. This yields a continuous changing of the schemata.

The problem is that such systems could be over-trained.

e.g.: Calligrapher mostly recognizes “g“ or “y“ when a user enters the letter “q“.

It follows from this that up to ten schemata for the letter “q” can exist. In successive use, the input will be recognised as a "q" more often than as the letters "g" and "y" due to the similarity of their schemata.

In the new version each letter is entered 3 times during the Calibration. The calibrated schemata are static. They don’t change unless the user makes a new Calibration.

The following characteristics of a letter are also saved:

- Number of used strokes
- If a letter is “high”(f, h, b, t, k, l) or “low” (q, g, y, j, p) or neither of them (a, e, s…)
- The “length” of the typed letter
- The direction at the start and at the end

With regard to the calibration process, there are a number of parameters that can be employed to influence the accuracy of the recognition:

Letter Combinations: In the previous version, the calibration was so concerted that each letter was typed twice. If Calligrapher does not recognize the right letter this letter was requested again and again (up to 10 times). The user was trained to give the input that matched the right result. After some time working with the device, the user knows how to write a letter in the appropriate manner to get the needed result. In the end, the user works only with Calligrapher because it already enters the letters correctly without no more need of the schemata..

Our new application follows another way. Each letter has to be entered three times. All three schemes of these entries are stored (unless the schemata are identical).

Since Calligrapher also matches words, it is possible that when typing a character, letter-combinations are also detected. (E.g. the character “a” can be recognized as “oi” or something similar)

In contrast to the previous version, this fact is taken into account. Letters may be better recognized as the distinctive features are larger.

Strokes: A further step in the refinement of character recognition is that the number of used strokes is also stored. An example of this would be that most users write a “q” with two strokes. In this case a “q” will no longer be mistaken for a “g” or a “y” for example.

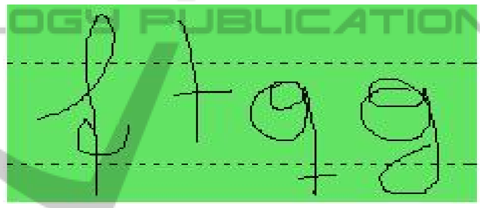

Figure 3: Distinction based on the number of strokes.

Letter Height: Not only the art of writing varies from user to user, the height of the writing can vary as well. While some users need the whole screen for a letter others could work with 1/3 of it. The user should have the possibility to not change his writing style only for better recognition.

During the calibration the highest point in the y-direction is detected. With the help of these points a “line” is drawn. That line differences the “low” letters (g, y, j, …) from the “high” ones (f, t, b, …). The lower line for letters such as q, y, g is visible to give an orientation for the preparation of the pen.

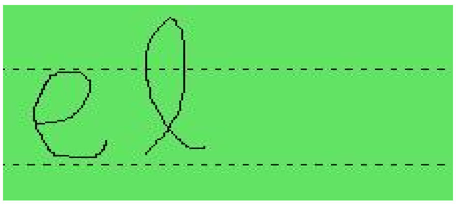

Figure 4: Distinction based on the height of the letter.

Note: The bottom line is already fixated. However, the top one is not- this means that it is calculated after the calibration.

Example:

Schema l:	*Schema e:*
*<Letter1Value="**e**" />*	*<Letter1Value="**e**" />*
*<Letter2Value="**l**" />*	*<Letter2Value="**l**" />*
<Letter3Value="" />	*<Letter3Value="**c**" />*
<Letter4Value="" />	*<Letter4Value="" />*
<Letter5Value="" />	*<Letter5Value="" />*
*<StrokesValue="**1**" />*	*<StrokesValue="**1**" />*
*<SmallCharacterValue="**1**" />*	*<SmallCharacterValue="**0**" />*

Note: As you can see the Results recognizes by Calligrapher are very similar. The 1 in SmallCharacter characterizes the "l" as a "height" letter.

Tape Length: The length of a character is measured and stored. This value is especially used for letters like "v" and "w", which are very similar to one another, whereas they contrast in their length.

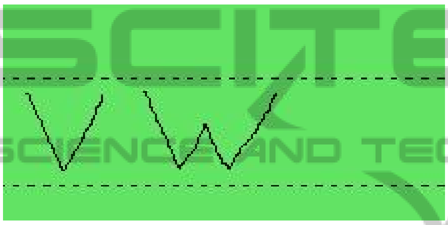

Figure 5: Distinction based on the tape length.

Example:

Letter v:	Letter w:
*<Letter1Value="**v**" />*	*<Letter1Value="**v**" />*
*<Letter2Value="**V**" />*	*<Letter2Value="**V**" />*
*<Letter3Value="**r**" />*	*<Letter3Value="**r**" />*
*<Letter4Value="**l**" />*	*<Letter4Value="**l**" />*
<Letter5Value="" />	*<Letter5Value="" />*
*<StrokesValue="**1**" />*	*<StrokesValue="**1**" />*
*<SmallCharacterValue="**0**" />*	*<SmallCharacterValue="**0**" />*
*<LengthLetterValue="**16**" />*	*<LengthLetterValue="**16**" />*

Note: The letter "w" was matched very badly from the Calligrapher. Only the tape length is an indicator for the recognition.

Direction Vector: This is the last criteria for letter recognition. The direction in which the user guides the AluPen when writing is stored. This helps with letters like v and r, for example. Most users make a curve down when writing an "r" and a curve up when writing a "v".

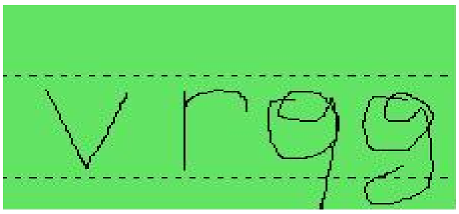

Figure 6: Distinction based on the direction vector.

Example:

Letter g:	Letter q:
*<EndVektYValue="**0**" />*	*<EndVektYValue="**1**" />*
*<EndVektYValue="**0**" />*	*<EndVektXValue="**0**" />*

4.5 Correction Intervention

This process describes the translation of a typed letter. Calligrapher is designed so that it does not only recognize letters but also words. For a typed letter, it provides a list with possible outcomes and the probabilities of their outcome.

Although Calligrapher provides a false result, the correct letter can be determined by using the correct schemata. The schemata are created during Calibration and stored in an xml-file. The result of an input character is compared with the schemata.

A letter has been recognized perfectly well when Calligrapher returns only one or two possible letters as the result. The input is also very clear when the result list does not contain any letter-combinations. In this case, the results are compared only to the matching schemata. The best fitting is taken as the result.

If the result list contains more than two results or if it contains letter combinations, the results are compared with all schemata.

The main challenge is to find out how the stored characteristics should be handled to get the correct result.

5 EMPIRICAL EVALUATION

The major difficulty is that every user writes differently. This makes it hard to cover all possibilities while giving the user complete freedom when writing.

Another large problem of recognition is encountered when a user only uses cursive letters. In this case, the strokes and the vectors do not bring any advantages – letters like r, v and q are more difficult to recognize.

A very famous system for character recognition is Graffiti. This recognition system uses unistrokes for each characters to enable error-free typing.

After the Calibration, our system checks the input of the user, and if there are various inputs of a single letter (strokes or height) the user is asked to enter it again. Even when they can choose their style of inputting, the user is "trained" to consistancy in their writing.

We conducted two test phases and after every stage we analyzed the data and tried to reduce the errors.

First Test Phase

The tests were made with 15 students from Graz University of Technology. Every test person had to do a calibration for both systems and to insert the German alphabet. The next test included the task to input the following sentence:

"Die heiße Zypernsonne quälte Max und Victoria ja böse auf dem Weg bis zur Küste".

This sentence contains every character of the German alphabet. Both systems achieved only about 70% accuracy.

The users writing in block letters were recognized well by both systems in contrast to cursive writing.

After this test phase, we began to search for reasons and made some fundamental changes.

Second Test Phase

In this phase, the line for the "high" letters was invisible. But most users began to write smaller than during the calibration. So we decided to make the line visible after the calibration for the second test phase. The test persons felt very comfortable with the lines because they gave them a starting point for the pen. Further, we found some errors in the character recognition which is based on the schemata of a user. We changed the whole process of result finding.

The character length and the vectors now only come into consideration when we have to decide between special characters (e.g. "v", "w", "n", "u", "r" or "q", "g", "y").

no.	no. char.	no. faults	no. recogn.	perc.
1	96	9	87	90,63
2	60	11	49	81,67
3	96	2	94	97,93
4	60	0	60	100,00
5	96	7	89	92,71
6	96	5	91	94,80
7	60	5	55	91,67
8	96	8	88	91,67
9	60	4	56	93,33
10	59	4	55	93,22
11	60	5	55	91,67
12	60	3	57	95,00
13	120	3	117	97,50
14	60	6	54	90,00
15	112	25	87	77,68
16	60	0	60	100,00
Total	**1251**	**97**	**1154**	**92,25**

Figure 7: The results of the second test-phase.

Further, we immediately look at all 3 schemata for each character in order to give the matching character-combinations preference. After these changes we made a second test phase with 15 users and achieved an accuracy rating of 92%.

As with any HWR-System, we had the option of "forcing" the user to a precious input (e.g. WritePad) or to adopting certain requirements for better recognition (e.g. Graffiti). We decided to take a step towards systems like Graffiti. Since we did not want to restrict users too much, we only focused on some problem letters.

Example:

"v" is mainly confused with "r", "u" or "w"
"q" with "g" or "y"
*"h" with "k" or "b"*These misinterpretations of course, depend on the user and their art of writing. Due to that fact, the problem letters could be determined during the calibration. With regard to these particular letters, the user would have to change their art of writing. A combination of the recognition with a built-in dictionary could improve the accuracy further.

Another important point is, that after some time working with the system in its current state (for example 1 week) the user can achieve an accuracy rating of 97% without problems.

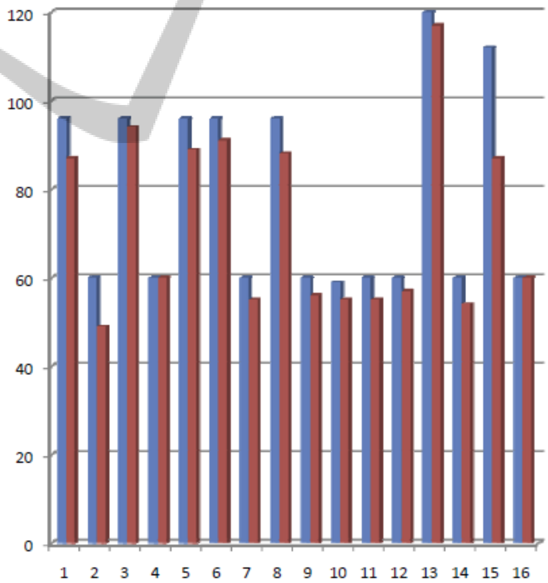

Figure 8: Visualization of the results obtained in the second test phase: no. words in red, no. of characters in blue.

Number Recognition: We also tested the number recognition. The users had to enter the actual date, time and all numbers from 0 – 9. In this case we achieved a accuracy rating of 95%.

The number recognition currently works without a schema. So it depends only on the results determined by Calligrapher.

6 CONCLUSIONS

Our main goal was to further improve the current system, along with getting insight into currently available systems.

With a much better recognition rate, we have achieved the primary goal, especially the letters from end users who mostly write cursive can be recognized much better.

However, the system still has the potential to be further refined. Based on our statistical tests, it is planned to further improve the handwriting recognition and to bring a system based on the studies presented in this paper and with a word recognition feature into the mobile phone market.

Additionally, this would help to reduce false results since the recognized letter must be within the context of a word.

Moreover, in order to enable a rapid entry, we are experimenting with speech recognition features, since natural language interaction is highly important, in addition to the handwriting recognition (refer also future outlook).

7 FUTURE OUTLOOK

Generally the interest in using handwriting recognition will rather drop in the future (c.f. with Steve Jobs "who needs a stylus") – although Apple has made a new patent application in handwriting and input recognition via pen (Yaeger, Fabrick and Pagallo, 2009)

The reason for not using a stylus is twofold:

1) the finger is an accepted natural input medium (Holzinger, 2003), and

2) touch-based computers have gained a tremendous market success.

In future, communication and interaction on the basis of Natural Language Processing (NLP) will become more important.

However, within the professional area of medicine and health care, stylus-based interaction is still a topic of interest, because medical professionals prefer, and are accustomed to the use of a pen, therefore a stylus (Holzinger et al., 2008b).

Consequently, research in that areas is still promising.

ACKNOWLEDGEMENTS

We thank Mr. Ferk, the CEO of the FERK company, for his support.

REFERENCES

Dzulkifli, M., Muhammad, F. & Razib, O. (2006) On-Line Cursive Handwriting Recognition: A Survey of Methods and Performance. *The 4th International Conference on Computer Science and Information Technology (CSIT2006).* Amman, Jordan 5-7 April, 2006.

Gader, P. D., Keller, J. M., Krishnapuram, R., Chiang, J. H. & Mohamed, M. A. (1997) Neural and fuzzy methods in handwriting recognition. *Computer,* 30, 2, 79-86.

Gowan, W. (2004), Optical Character Recognition using Fuzzy Logic. Online available: http://www.freescale.com/files/microcontrollers/doc/app_note/AN1220_D.pdf, last access: 2011-02-18

Graves, A. & Schmidhuber, J. (2009), *Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks.* Online available: http://www.idsia.ch/~juergen/nips2009.pdf, last access: 2011-02-17

Holzinger, A. (2003) Finger Instead of Mouse: Touch Screens as a means of enhancing Universal Access. In: Carbonell, N. & Stephanidis, C. (Eds.) *Universal Access: Theoretical Perspectives, Practice and Experience, Lecture Notes in Computer Science (LNCS 2615)* Berlin, Heidelberg, New York, Springer, 387-397.

Holzinger, A., Geierhofer, R. & Searle, G. (2006) 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*. München, Oldenbourg, 339-347.

Holzinger, A., Hoeller, M., Bloice, M. & Urlesberger, B. (2008a). Typical Problems with developing mobile applications for health care: Some lessons learned from developing user-centered mobile applications in a hospital environment. International Conference on *E-Business (ICE-B 2008)*, Porto (PT), IEEE, 235-240.

Holzinger, A., Höller, M., Schedlbauer, M. & Urlesberger, B. (2008b). An Investigation of Finger versus Stylus Input in Medical Scenarios. *ITI 2008: 30th International Conference on Information Technology Interfaces,* Cavtat, Dubrovnik, IEEE, 433-438.

Holzinger, A., Schlögl, M., Peischl, B. & Debevc, M. (2010) Preferences of Handwriting Recognition on Mobile Information Systems in Medicine: Improving handwriting algorithm on the basis of real-life usability research (Best Paper Award). *ICE-B 2010 - ICETE The International Joint Conference on e-Business and Telecommunications.* Athens (Greece), INSTICC.

Lee, S. W. (1999), Advances in Handwriting Recogntion (*Series in Machine Perception and Artificial Intelligence.* Online available, last access:

Liu, Z., Cai, J. & Buse, R. (2003) *Handwriting Recognition: Soft Computing and Probabilistic Approaches.* New York, Springer.

Perwej, Y. & Chaturvedi, A. (2011) Machine recognition of Hand written Characters using neural networks.

International Journal of Computer Applications, 14, 2, 6-9.

Phatware (2008) *Calligrapher SDK 6.0 Developer's Manual.*

Pittman, J. A. (2007) Handwriting Recognition: Tablet PC Text Input. *IEEE Computer,* 40, 9, 49-54.

Plamondon, R. & Srihari, S. N. (2000) On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 22, 1, 63-84.

Plotz, T. & Fink, G. A. (2009) Markov models for offline handwriting recognition: a survey. *International Journal on Document Analysis and Recognition,* 12, 4, 269-298.

Shu, H. (1997), *On-Line Handwriting Recognition Using Hidden Markov Models.* Online available: http://dspace.mit.edu/bitstream/handle/1721.1/42603/37145316.pdf, last access: 2011-02-18

Sulong, G., Rehman, A. & Saba, T. (2010) Improved Offline Connected Script Recognition Based on Hybrid Strategy. *International Journal of Engineering Science and Technology,* 2, 6, 1603-1611.

VisionObjects (2009), *MyScript Stylus.* Online available: http://www.visionobjects.com/handwriting_recognition/DS_MyScript_Stylus_3.0.pdf, last access: 2011-02-15

Wang, F. & Ren, X. S. (2009) *Empirical Evaluation for Finger Input Properties In Multi-touch Interaction.* New York, Assoc Computing Machinery.

Willis, N. (2007), *CellWriter: Open source handwriting recognition for Linux*, Online: . Online available: http://www.linux.com/archive/feed/120867, last access: 2011-02-18

Yaeger, L. S., Fabrick, R. W. & Pagallo, G. M. (2009) *Method and Apparatus for Acquiring and Organizing Ink Information in Pen-Aware Computer Systems* 20090279783.

Zafar, M. F., Mohamad, D. & Othman, R. (2006) Neural Nets for On-line Isolated Handwritten Character Recognition: A Comparative Study. *The IEEE International Conference on Engineering of Intelligent Systems (ICEIS 2006).* Islamabad, 22-23 April 2006.

Zafar, M. F., Mohamad, D. & Othman, R. M. (2005) On-line Handwritten Character Recognition: An Implementation of Counterpropagation Neural Net. *Journal of the Academy of Science, Engineering and Technology (Available online: http://www.waset.org/journals/waset/v10/v10-44.pdf),* 10, 232-237.