Seamless Integration of Handwriting Recognition into Pen-Enabled Displays for Fast User Interaction

Marcus Liwicki*, Markus Weber*†, Tobias Zimmermann*†, Andreas Dengel*†
* German Research Center for AI (DFKI)
Kaiserslautern, Germany
{firstname.lastname}@dfki.de
† Department of Computer Science, University of Kaiserslautern,
P.O. Box 3049, 67653 Kaiserslautern,Germany

***Abstract*—This paper proposes a framework for the integration of handwriting recognition into natural user interfaces. As more and more pen-enabled touch displays are available, we make use of the distinction between touch actions and pen actions. Furthermore, we apply a recently introduced mode detection approach to distinguish between handwritten strokes and graphics drawn with the pen. These ideas are implemented in the *Touch & Write* SDK which can be used for various applications. In order to evaluate the effectiveness of our approach, we have conducted experiments for an annotation scenario. We asked several users to mark and label several objects in videos. We have measured the labeling time when using our novel user interaction system and compared it to the time needed when using common labeling tools. Furthermore, we compare our handwritten input paradigm to other existing systems. It turns out that the annotation is performed much faster when using our method and the user experience is also much better.**

Index Terms*—Handwriting recognition; Applications; HCI; Mode detection; *Touch & Write

I. Introduction

Handwriting recognition (HWR) has been the topic of research for many decades. While the first recognizers have been developed for isolated characters or digits, later recognizers focused on complete words or even sentences [1], [2], [3]. Even after several decades of research, the task of handwriting recognition cannot be assumed to be solved already. There is still room for improvement of the recognition performance, as well as handling different scripts and special environments. [4].

Despite the open issues in HWR, nowadays there exist solutions which have a quite good recognition performance (e.g., recognizers from Microsoft© and Vision Objects©). Both mentioned recognition systems are applicable on online data, i.e., the time sequence of the points is available. This data is usually acquired using a specific digital pen. Note that the more difficult case, i.e., offline handwriting is not a topic of this paper since our main focus is on direct pen input for natural user interfaces in real-time environments.

However, we can observe that handwriting as an input metaphor is only rarely used for electronic systems in our daily life. In fact, even the most prominent touchable display, the iPad, does not come with integrated handwriting recognition; and the Tablet PCs from Microsoft include handwriting recognition already since Windows XP but still it is not used.[1] The main problems are not the performance of the recognition systems themselves, but the way how the handwriting is integrated into the whole process. If switching between several modes (handwriting vs. selecting) or opening separate input dialogs is required, the acceptance of the system drops dramatically.

Fig. 1: Handwriting input just where it is needed

In this paper we propose a solution for the problem of less acceptance. By including several state-of-the-art recognition techniques and using modern input devices, we are able to design intuitive interfaces for natural handwritten input. In particular, we developed the *Touch & Write* SDK for creating applications with integrated multi-touch and pen input. This novel framework can be applied for developing several handwritten input applications.

An example illustration of our proposed idea is given in Figure 1. There, an object appearing is annotated for later use. The user can simply write the annotation at the desired place and after recognition, the video frame is annotated at that position.

The rest of the paper is structured as follows. First, Section II motivates the usage of handwriting as input metaphor. Next, Section III describes our approach of integrating handwriting into practical systems in a natural way. Subsequently, Section IV reports on experiments performed in order to show the practicability of our approach. Finally, Section V draws some conclusions.

[1]The Microsoft Windows XP Tablet PC Edition SDK is available for download at http://www.microsoft.com/download/en/details.aspx?id=20039

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.78

IEEE computer society

II. Why using Handwritten Input

A standard reaction of computer-affine people to this question is that they are much faster with a keyboard. And it is true, handwriting as an input metaphor is a very slow interaction process when you are trying to input information into the computer, especially for those people who learned typing rapidly with the keyboard.

However, if we just look back around 150 years[2], people mainly used a pen for putting down information on paper. Even today and even on computer science related conferences and meetings we can see many people taking notes with a pen. The pen is always available, it is a cheap device and we do not need to take care about how to achieve a desired visual appearance, we can just concentrate on the content. Not being concerned about "how" but just concentrating on "what" to write improves the outcome of creativity sessions, because we are not disturbed by interaction barriers and therefore do not need to interrupt our thinking process. Recent experimental studies support the general opinion that just the process of handwriting improves the creativity, especially for proficient writers [5]. A possible explanation is that during writing a huge number of fine muscle motor plans are involved which activate the brain.

The use of buttons on the other hand side was invented for the interaction with machines, because they did not understand other means of communication. By introducing the keyboard the human adapted to the machine and not vice versa. This resulted in problems like restrictions to a finite set of symbols which can be easily put down, losing the variety of handwritings and the individuality of a person's handwriting, and unhealthy ways of finger and hand movements leading to typist's cramp. Not to forget the introduction of on-screen keyboards which have been invented for touch interfaces: Especially on small screens like mobile devices it is difficult to hit the correct keys and the process of generating letters slows down dramatically. Even the mouse does not completely reflect the interaction paradigms of the real world, as it just offers an indirect 2D navigation. Studies in [6] and [7] have shown that especially elderly people have problems to interact with computers using a mouse - pens have proven to improve the usability.

As the performance of online handwriting recognition systems nowadays is very high [8] and the systems work almost perfectly in restricted domains, handwritten input should be integrated into several application scenarios. Instead of using complicated metaphors like mode switches and dialogs we suggest using specific hardware to automatically distinguish between pen and touch interactions, as described in the next section.

III. The *Touch & Write* User Interface and SDK

Usability is the main aim of Human-Computer Interaction (HCI) research. It is essential to design interfaces that allow users to intuitively work and interact with applications even for first time users. Appropriate metaphors and devices have to be used to allow easy and fast interaction. The new generation of intuitive devices seems to be found in Multi-Touch (MT) environments, which provide hands on experience and offer a wider domain of usage scenarios. However, current MT solutions, such as the Microsoft Surface[3] or DiamondTouch [9], lack in a way of intuitively switching between two important modes: moving the objects and editing their content, i.e., drawing or writing.

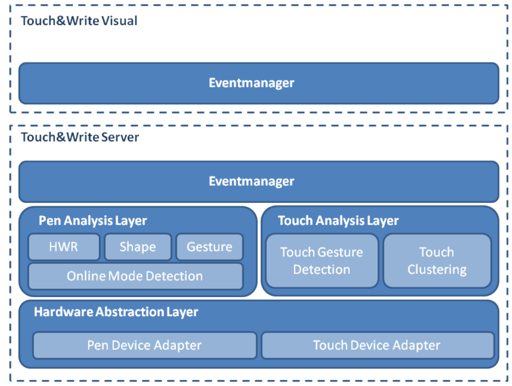

Fig. 2: *Touch & Write* SDK overview

To cope with this problem, we have recently proposed a tabletop solution which also integrates pen input. The *Touch & Write* Table [10] integrates the Anoto technology[4] on top of the usual frustrated total internal reflection (FTIR) proposed by Han [11]. This table was used for the work presented in this paper, however, a mobile solution is also available, which runs on a small pen-enabled tablet PC.

Recently, we developed an SDK for touch tables integrating the novel input device, the pen (like the *Touch & Write* table). The *Touch & Write* SDK is an approach to simplify the development for a tabletop device with pen-support. Figure 2 illustrates the components of the *Touch & Write* SDK. The touch and pen actions are processed separately. Since the focus of this paper is the handwritten input, this component is discussed briefly. More details on the SDK in general can be found in [12].

While the user interacts with the system the pen data is cached and whenever the user stops to interact for a short interval T_{delay} an on-line mode detection is triggered [13]. Depending on the detection result, further analysis is applied on the cached on-line pen data. Either the handwriting recognition is used to detect written text, the shape detection subsystem analyzes the drawing and recognizes geometrical shapes, or the gesture recognizer is used to recognize a gesture. This allows developing easy-to-use applications, because no explicit mode-switch is necessary and direct feedback is given.

[2] The first commercial typewriter was produced in 1865

[3] http://www.microsoft.com/surface: Last accessed 05/22/2010

[4] Anoto pen http://www.anoto.com/: Last accessed 20/10/2011

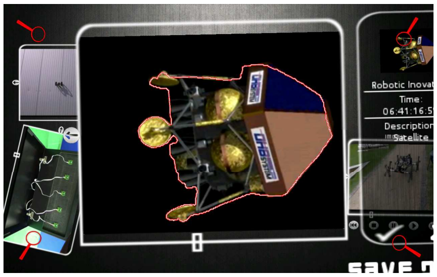

Fig. 3: Outlining an object for annotation

IV. Evaluation

In order to evaluate the practicability of our system we use a scenario where the usage of handwritten and pen-input is needed. Annotating videos is such a task and it has a great benefit for several communities, as labeled ground truth data is the foundation for supervised machine learning approaches. Thus, there is need for an easy-to-use tool which assists users with labeling even complex structures. For outlining the shape and labeling an object, we developed an application based on the *Touch & Write* SDK. The application is named *CoVidA* for Collaborative Video Annotation tool. This will be briefly described in the following section.

A. CoVidA *for Video Annotation*

Most of the current tools support mouse and keyboard input, such as ANVIL [14], *LabelMe* [15], [16] or LIGVID [17]. However, while a mouse is sufficient for outlining objects with a simple shape, it has major drawbacks when it comes to more complex shapes. Hence, using a pen device which directly interacts with the display seems to be a promising approach.

As stated above, *CoVidA* makes use of multi-touch and pen interaction. While the touch gestures are used to manipulate and interact within videos and other visual components, the pen is used for annotating and outlining desired objects in videos. Furthermore, it can be used for querying for already written annotations. Similar to a real pen, the digital-pen is used for outlining the desired shapes and for writing the annotation terms.

If the user completes outlining an object in the image, the online mode detection detects a shape drawing. Then this shape is send to the *CoVidA* application. Thus it is assumed that the user aims for annotating this object (see Figure 3). Afterwards, the user can write some text on the object and the detected handwritten text is added as an annotation to the outlined object. By having this kind of automatism, users are not forced to switch the current mode manually.

B. *Evaluation Goals*

The evaluation aims at three main goals:

1) Annotation of complex structures using a pen device.
2) Investigate the ease of interaction with video data in *CoVidA*.
3) Possibility to work with *CoVidA* without detailed explanation.

In order to compare our application to other metaphors, we installed the *CoVidA* software on the *Touch & Write* table and the *LabelMe* application was started in an Internet browser on the standard desktop PC, as well as on a Tablet PC.

C. *Participants*

17 volunteers participated in our study. We separated them into groups with a different skill level, one group with no experience with this kind of software and users working in an area where annotation tasks are quite common. Both groups of participants had the same experimental settings, no compensation for different skills was offered.

1) Group 1: consists of 14 volunteers (8 male and 6 female), aged between 17 and 28. All participants of Group 1 use the personal computer frequently (with mouse and keyboard input) and had no experience with combined touch and pen input.

2) Group 2: consists of 3 volunteers (3 male), aged between 28 and 29. The participants of Group 2 have expert knowledge in conventional annotation software (with mouse and keyboard input). One of them is a designer who regularly uses Wacom pen tablets.

D. *Task*

In order to evaluate the effectiveness of a pen device as input metaphor for annotation tasks, we asked each participant to perform three times two tasks with an increasing complexity level. We defined an outlining task for objects in a video frame and a labeling task for these objects. The outlining task is divided into three different complexity levels determined by the complexity of the shape. In summary, the tasks are:

- Outlining:
  - *CoVidA* : The test person draws along the shape of the object.
  - *LabelMe* : The test person clicks with the mouse along the shape of the object.
  - *LabelMe* on Tablet-PC: The test person taps with the pen along the shape of the object.
- Labeling:
  - *CoVidA* : The test person writes using the pen the name of the object on the video.
  - *LabelMe* : The test person writes using the keyboard the name of the object in the text field.
  - *LabelMe* on Tablet-PC: The test person opens the dialog for writing, writes using the pen the name of the object in the text field and confirms the writing process.

To evaluate the effectiveness of the systems we measured the time needed to outline the shape and the time needed label the object independently. Note that in many cases it is important to have an accurate outline of the objects in the image/video frame, thus the participants had to outline the shape of the

(a) Goal (b) Person. (c) Satellite.

Fig. 4: Video frames used for the annotation task

object as close as possible. We used the following complexity levels for objects:

1) At this level the participants had to annotate a soccer goal (see Figure 4a) with *CoVidA* and *LabelMe* in the same video frame. This experiment aims on measuring how fast the user can annotate simple objects (only four lines) with a simple description.
2) Here the objective was to annotate a person (see Figure 4b) with *CoVidA* and *LabelMe* . The aim was to measure how fast a user could accomplish an annotation of a more extensive object. The annotation of a person is a common task in building a ground truth for surveillance or person tracking software.
3) The last objective includes the measurement how efficient the participant could outline a complex structure, such as a satellite (see Figure 4b) with the label satellite.

E. Procedure

All participants have been shortly introduced to the functionality of *LabelMe* and *CoVidA* . Each participant performed the tasks separately in the room with the test setup. Now, each task had been explained by the test operator, and the participant accomplished the task independently. The time duration for each task was measured by an independent test operator, Tables I and II show the average time needed for each task. After the completion of all tasks with *CoVidA* and *LabelMe* the test operator asked ten questions taken from [18].

Furthermore, each participant filled out a questionnaire. From these questionnaires the System Usability Scale (SUS) has been calculated (see second column in Table III for Group 1 and Table IV for Group 2, respectively).

F. Results

The results of the user study show that all participants required more than twice the time on *LabelMe* than on *CoVidA* for more complex structures. For the simple shape, there was no significant difference between both applications. As most of the users are frequently using keyboards, we observed a better performance with typing the label for the objects. Thus for users with experience with the usage of keyboards, it is still more efficient to use the keyboard than using the pen for writing the labels. However, because of the saving in time for the outlining task (see Figure I and Figure II), *CoVidA* still

TABLE I: Task Results Group 1

Application	Task	Time in s		
		outlining	labeling	sum
CoVidA	1	5,23 ± 1,48	3,72 ± 1,33	8,95 ± 2,11
	2	8,47 ± 4,06	4,79 ± 2,07	13,26 ± 4,92
	3	14,14 ± 5,16	5,48 ± 1,73	19,62 ± 5,86
LabelMe	1	6,81 ± 1,78	3,56 ± 1,02	10,37 ± 2,24
	2	33,25 ± 10,96	3,53 ± 0,87	36,78 ± 10,73
	3	57,79 ± 16,03	4,2 ± 1,52	61,99 ± 15,36
LabelMe	1	5,34 ± 1,24	9,36 ± 2,80	14,70 ± 3,13
+Microsoft	2	21,08 ± 6,45	10,40 ± 2,81	31,48 ± 7,99
HWR	3	35,94 ± 11,81	10,29 ± 2,12	46,22 ± 13,34

TABLE II: Task Results Group 2

Application	Task	Time in s		
		outlining	labeling	sum
CoVidA	1	4,4 ± 2,06	3,05 ± 0,72	7,45 ± 2,71
	2	8,89 ± 3,44	3,40 ± 0,84	12,28 ± 4,27
	3	21,37 ± 16,12	5,27 ± 0,99	26,63 ± 16,83
LabelMe	1	6,28 ± 1,21	3,49 ± 0,30	9,76 ± 1,43
	2	18,41 ± 2,20	4,48 ± 2,87	22,89 ± 5,06
	3	46,89 ± 6,15	2,94 ± 0,57	49,83 ± 6,64

Fig. 5: Handwriting input for the Microsoft Tablet PC

has shown to be more efficient than *LabelMe* for the complete annotation task. Note that the overall annotation time can again increase if a switch between using a pen and using a keyboard would have to be performed.

The experiments using *LabelMe* on a Tablet PC have only been performed by Group 1 since the annotation time and the outlining time was significantly longer than using the *CoVidA* tool. The participants did not feel comfortable using this dialog-based input for handwriting. Just for writing the labels, first a handwriting input dialog needs to be opened and then the text can be written. In general, the realization of handwritten input is even more complicated. As depicted in Fig. 5 the user usually has to navigate through several

TABLE III: Questioning Results Group 1

Application	SuS Score [18]
CoVidA	86,15
LabelMe	75,19

TABLE IV: Questioning Results Group 2

Application	SuS Score [18]
CoVidA	91,67
LabelMe	75,83

dialogs in order to correct recognition mistakes. This is a very cumbersome process. However, in our experiments we asked the participants not to care about the recognition result even if it was wrong. Still some users were distracted by the intermediate recognition results displayed in real-time. It turned out that a little delay is actually desired.

Noteworthy, even the outlining task was slower when using the *LabelMe* interface. This can be explained by the fact that drawing comprises a more natural way of connecting points than always lifting the pen between two consecutive points.

A further important advantage of the *CoVidA* system is that the accuracy of the border was much better when directly drawing with the pen compared to the point-to-point interaction.

The users opinion to the usability of *CoVidA* and *LabelMe* are throughout positive. Nevertheless, the participants preferred *CoVidA* because the interaction using the combination of multi-touch and a pen was perceived as more intuitive and easy to use. Especially during outlining more complex objects they preferred the direct manipulation using the pen device.

V. Conclusions

In this paper we have proposed a framework for the integration of handwriting recognition into natural user interfaces. Our main claim is that handwriting and should be performed using a pen and there should be no explicit mode switch between several pen input modes. We described the *Touch & Write* table and SDK as a successful implementation of this framework.

Furthermore, we performed experiments measuring the effectiveness of the novel input metaphors. We observed both, a speed up of annotation tasks as well as an improved user experience even for users having expert knowledge in conventional annotation software (with mouse and keyboard input). Noteworthy, the handwritten text was recognized correctly at almost all cases.

In summary we have shown that handwritten input can be seamlessly integrated into applications in order to improve the effectiveness and the user experience in several application areas. Since the handwriting recognition has also shown a good performance (especially if the domain is known) we suggest to develop more intuitive applications with handwritten input. Our aim is that with the increasing popularity of direct touch interfaces, the metaphor of handwritten interaction will also become more prominent, especially, if the recognition is seamlessly integrated into the applications as it is done in *CoVidA* .

Acknowledgment

This work was funded by the German Federal Ministry of Education and Research (BMBF) in the EMERGENT project and the ADIWA project.

References

[1] H. Bunke, "Recognition of cursive Roman handwriting – past present and future," in *Proc. 7th Int. Conf. on Document Analysis and Recognition*, vol. 1, 2003, pp. 448–459.

[2] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: a comprehensive survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[3] A. Vinciarelli, "A survey on off-line cursive script recognition," *Pattern Recognition*, vol. 35, no. 7, pp. 1433–1446, 2002.

[4] *12th Int. Conf. on Frontiers in Handwriting Recognition*, 2010.

[5] J. R. Hayes and V. Berninger, "Relationships between idea generation and transcription: How act of writing shapes what children write," 2010.

[6] N. Charness, E. A. Bosman, and R. G. Elliott, "Senior-Friendly Input Devices: Is the Pen Mightier than the Mouse?" in *103rd Annual Convention of the American Psychological Association Meeting*, New York, New York, Aug. 1995.

[7] M. W. Smith, J. Sharit, and S. J. Czaja, "Aging, motor control, and the performance of computer mouse tasks." *Human factors*, vol. 41, no. 3, pp. 389–96, Sep. 1999. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10665207

[8] M. Liwicki and H. Bunke, "Combining diverse on-line and off-line systems for handwritten text line recognition," *Pattern Recognition*, vol. 42, no. 12, pp. 3254–3263, 2009.

[9] P. Dietz and D. Leigh, "Diamondtouch: a multi-user touch technology," in *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM, 2001, pp. 219–226.

[10] M. Liwicki, O. Rostanin, S. M. El-Neklawy, and A. Dengel, "Touch & write: a multi-touch table with pen-input," in *9th Int. Workshop on Document Analysis Systems*, 2010, pp. 479–484.

[11] J. Y. Han, "Low-cost multi-touch sensing through frustrated total internal reflection," in *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM, 2005, pp. 115–118.

[12] M. Liwicki, M. Weber, and A. Dengel, "Online mode detection for pen-enabled multi-touch interfaces," in *Proc. 15th Conf. of the International Graphonomics Society*, 2011, p. 4 pages.

[13] M. Weber, M. Liwicki, Y. T. H. Schelske, C. Schoelzel, F. Strauss, and A. Dengel, "Mcs for online mode detection: Evaluation on pen-enabled multi-touch interfaces," in *Proc. 11th Int. Conf. on Document Analysis and Recognition*, 2011.

[14] M. Kipp, "Spatiotemporal coding in anvil," in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association, 2008.

[15] B. Russell, A. Torralba, and K. Murphy, "LabelMe: a database and web-based tool for image annotation," *International Journal of*, vol. 77, no. 1-3, pp. 157–173, Oct. 2008. [Online]. Available: http://www.springerlink.com/index/76X9J562653K0378.pdf

[16] J. Yuen, B. Russell, and A. Torralba, "LabelMe video: Building a video database with human annotations," *2009 IEEE 12th International Conference on Computer Vision*, pp. 1451–1458, 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5459289

[17] S. Ayache, G. Quénot, and A. Tseng, "The ligvid system for video retrieval and concept annotation," in *Proceedings of the international conference on Multimedia information retrieval*, ser. MIR '10. New York, NY, USA: ACM, 2010, pp. 385–388. [Online]. Available: http://doi.acm.org/10.1145/1743384.1743447

[18] J. Brooke, "SUS: A quick and dirty usability scale," in *Usability evaluation in industry*, P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, Eds. London: Taylor and Francis, 1996.