Touch & Write — Penabled Collaborative Intelligence —

Andreas Dengel[1,2], Marcus Liwicki[1], and Markus Weber[1,2]

[1] German Research Center for AI (DFKI GmbH), Trippstadter Straße 122, 67663 Kaiserslautern, Germany,
`firstname.lastname@dfki.de`,
WWW home page: `http://www.dfki.de/~lastname`
[2] Knowledge-Based Systems Group, Department of Computer Science, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern

Abstract. Multi-touch (MT) technology becomes more and more famous and several frameworks have been proposed in the last decade. All of them focus on the support of touch input, gestures, and objects. Recently, however, a new generation of MT-tables emerged, which allows for pen-input in addition to the touch paradigm. These devices, such as the *Touch & Write* Table of DFKI, consider multiple pen interaction. In this paper we propose a software development kit (SDK) which integrates the basic processing of the pen input. Moreover, the *Touch & Write* SDK includes handwriting recognition and geometric shape detection. Using writing mode detection the SDK automatically applies the correct recognition component based on features extracted from the pen data.

Keywords: Software architecture, SDK, Touch & Write, sketch-based interface

1 Introduction

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.

Almost three decades after the computer mouse started its triumph with the Apple Macintosh, it is about time for a next generation of HCI. The mouse does not completely reflect the interactions paradigms of the real world. This new generation seems to be found in MT tabletop 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 [5], lack in a way of intuitively switching between two important modes: moving the objects and editing their content, i.e., drawing or writing.

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

To cope with this problem, we have recently proposed a tabletop solution which also integrates pen input. The *Touch & Write* Table [13] integrates the Anoto technology[4] on top of the usual frustrated total internal reflection (FTIR) proposed by Han [8]. Another table integrating touch and pen input was proposed in the Flux project [12], where a similar hardware is used.

In this paper we propose an SDK which enables developers to use touch and write input metaphors without the need of in-depth knowledge about pattern recognition and handwriting recognition. All these techniques are integrated into the framework and the developer just has to handle events which are triggered by a separate engine.

The rest of this paper is organized as follows. First, Section 2 presents the *Touch & Write* SDK with more detail. Next, Section 3 introduces an application which makes use of the SDK and allows architects to annotate and retrieve architectural floor plans. Subsequently, Section 4 introduces an application for shopping in smart environments and Section 5 presents an application for video annotation and retrieval. Finally, Section 6 concludes this paper and draws some outlook for future work.

2 *Touch & Write* SDK

2.1 Previous Frameworks

Using sketch-based interface is a promising approach in several domains as the usage of pen gives a user more freedom and a precise tool. Examples for sketch-based systems are COMIC system [14], Simulink's[5] Sim-U-Sketch [10], and recently an application to draw UML diagrams [7]. Unfortunately, most of the existing frameworks for multi-touch tables focus only on touch and object support.

MT4j [11] is designed for rapid development of graphically rich applications on a variety of contemporary hardware, from common PCs and notebooks to large-scale ambient displays, as well as different operating systems. The framework has a special focus on making multi-touch software development easier and more efficient.

Echtler introduced libTisch [6], a layered architecture which allows easy integration of existing software, as several alternative implementations for each layer can co-exist.

pyMT [1] is based on the Python programming language, thus it is portable and offers a broad framework. It comes with native support for many multi-touch input devices, a growing library of multi-touch aware widgets, hardware accelerated OpenGL drawing, and an architecture that is designed to let you

[4] Anoto pen http://www.anoto.com/: Last accessed 04/02/2010

[5] Simulink is an environment for multi-domain simulation and Model-Based Design for dynamic and embedded systems. http://www.mathworks.com/products/simulink/: Last accessed 06/20/2010

Fig. 1. Picture of the Touch&Write table

focus on building custom and highly interactive applications as quickly and easily as possible.

Microsoft Windows 7 (Windows Presentation Foundation) offers a comprehensive API for multi-touch support. As the framework is based on Microsoft Windows technology cross-platform portability is of course an issue. Similarly, Adobe Flash/Air also has begun to integrate multi-touch support, but here, there are severe performance issues, as full screen updates are notoriously performance intensive in Flash. Further releases will bring some improvements.

However, none of the above mentioned frameworks offers pen support and further analysis of the pen data. We propose the integration of pen devices and pattern recognition techniques.

Our research is mainly motivated by the fact that in several domains, such as design or engineering, sketches are used to exchange ideas in a collaborative environment. Therefore supporting pen devices is a promising approach. A pen offers more freedom for designers, they can just draw as on usual paper. Furthermore it does not limit their creativity, since the designer does not need to think about how he or she puts the information on the paper, there is no need to go through various context menus and dialogs. Unfortunately, if the information is just drawn on normal paper it is hard to transfer it back to the digital world.

The main idea of the *Touch & Write* table and SDK is to offer an innovative new platform for creating applications that users find natural to use. The table seamlessly integrates the paper world into the digital world. Editing, arranging and writing tasks can be easily performed by using hands and the pen. The SDK goes beyond traditional MT-SDKs by including automated pen-input analyses and interpretation.

2.2 *Touch & Write* Components

The *Touch & Write* table, illustrated in Figure 1, combines the paradigm of multi-touch input devices and a multi-pen input device. The table is a novel rear-projection tabletop which detects touching by using FTIR and a high resolution

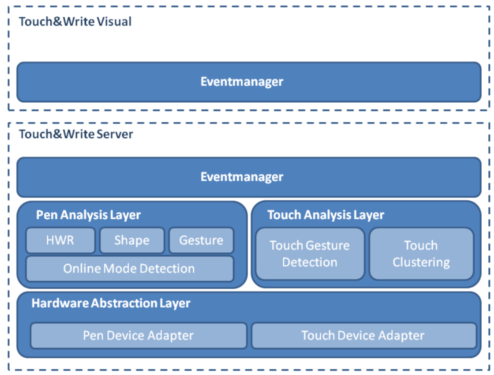

Fig. 2. *Touch & Write* SDK overview

pen technology offered by Anoto Group AB. This section shortly summarizes the main technical setup of the table surface.

Although touch and write interaction is very useful for several collaborative designing tasks, to the authors' knowledge there is no SDK available. Therefore we propose an SDK which integrates the novel input device, the pen. 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 & Write* SDK is divided into two parts, the *Touch & Write* Server and the *Touch & Write* Visual system. The basic idea is that the server part is always running on the system as service and handles the input devices. In contrast to the server component, the visual component is intended to be replaceable. Furthermore it should not be bound to any specific programming language. Therefore we use inter-process communication between the subsystems via a network protocol on the local machine. Each application using an implementation of the *Touch & Write* Visual component, has to register at the Event Manager of the server component.

The first layer of the server architecture is a hardware abstraction layer that offers adapters for the touch and pen input device. In the current development phase just the hardware of the *Touch & Write* table is supported. For the touch input the open source software Community Core Vision (CCV)[6] is employed which uses the TUIO protocol [9] and the on-line pen data is send via bluetooth. The adapter analyzes the streamed pen data and the coordinates are handed to the next layer.

The second layer is split up in two parts, the Pen Analysis Layer and Touch Analysis Layer. In this paper we focus on the pen data analysis, since the touch analysis can be done like in the above mentioned references. While the user

[6] http://ccv.nuigroup.com/: Last accessed 05/22/2010

interacts with the system the pen data is cached and whenever the user stops to interact for a short interval T_{delay} the on-line mode detection is triggered. 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 analysis the drawing and recognizes geometrical shapes, or the gesture recognizer is used to recognize a gesture.

Finally, the Event Manager will be informed about the result of the analysis. There is a general distinction between so called Low-Level Events and High-Level Events. Low-Level Events directly transmit the raw data, i.e., the screen coordinates extracted from the on-line pen data and the touch positions received CCV. High-Level Events contain the results of the analysis component, e.g., handwriting recognition results, detected shapes, or recognized gestures.

The event manager of the server component is responsible for distributing the events to the registered applications. Low-Level Events have to be delivered in real-time, giving the user immediate feedback, such as rendering the pen trajectory or for interaction with touched objects. While High-Level-Events are fired whenever the user stops the interaction with the system and the analysis is triggered.

As mentioned above, the visual component of the SDK is programming language independent. Each component needs to implement the events, and to receive them from server event manager by registering. Currently, there exist two implementations for the visual component. The first is based on a Java-based game engine called jMonkeyEngine[7]. There we have extended the jMonkeyEngine's input framework to seamlessly integrated pen and touch input. The second implementation is based on Adobe FLEX. An example how this works is shown at http://www.touchandwrite.de/.

3 a.Scatch - Sketch-Based Retrieval for Architectural Floor Plans

Recently, we have developed a system for architectural floor plan retrieval which builds on the *Touch & Write* SDK. It is a sketch-based approach to query a floor plan repository of an architect. The architect searches for semantically similar floor plans just by drawing a new plan. An algorithm extracts the semantic structure sketched on the *Touch & Write* table and compares the structure of the sketch with the ones from the floor plan repository (see Fig. 3). More details on this application can be found in [16].

Experiments investigating the performance of the Floor Plan Detection revealed that our approach outperforms most existing approaches [3, 2]. The detection rate on 80 images was more than 90% in many cases all plans were correctly recognized. Also the retrieval performance is close to 100% and it works very fast [16].

[7] http://www.jmonkeyengine.com/: Last accessed 05/22/2010

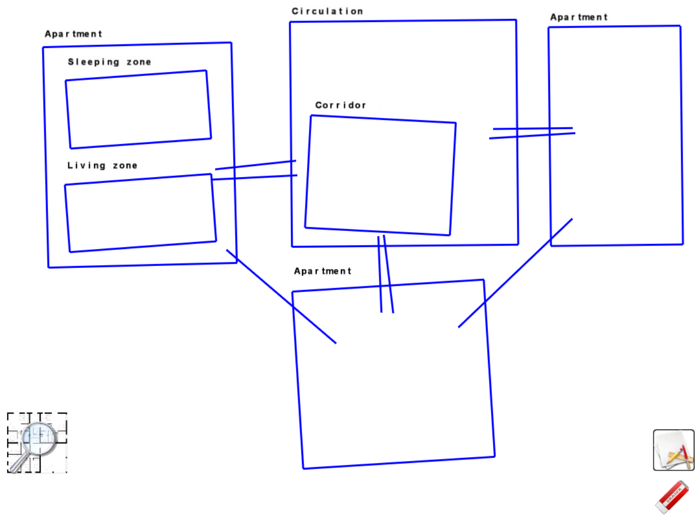

Fig. 3. a.SCatch retrieval interface

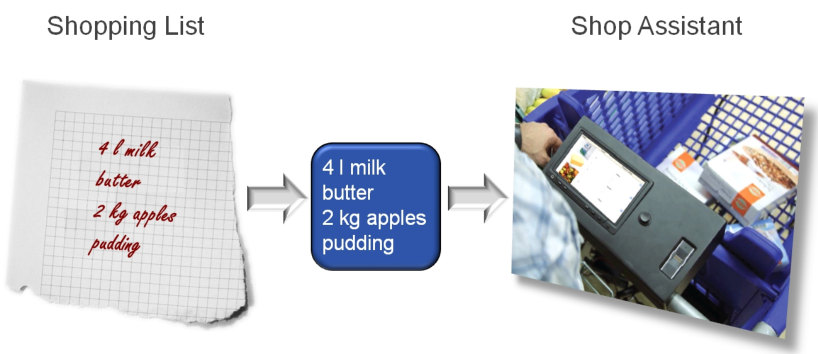

Fig. 4. SmartSL system overview

4 SmartSL - The Smart Shopping List

Besides the system for floor plan retrieval, we recently proposed a novel system which automatically extracts the intended items to buy from a handwritten shopping list. This intelligent shopping list relies on an ontology of the products which is provided by the shopping mall.

The work flow of the SmartSL system is illustrated in Fig. 4. The handwritten strokes are recognized by *Touch & Write* SDK components and then parsed in order to detect the amount and the desired item. This is then matched to the underlying ontology and the intended order is recognized. Finally, the user can use a specially equipped shopping cart to find the project in the marked and not to forget any item. Such carts exist in the Innovative Retail Laboratory (IRL) [15] located in St. Wendel. The IRL, a research laboratory of DFKI in cooperation with GLOBUS SB-Warenhaus Holding, illustrates how a shopping center in the future might look like.

Our current prototype works on an ontology of 300 products. In our real-world experiments we asked 20 persons to write shopping lists without any constrains. We have found out that in most cases the participants were happy with the retrieved results.

5 CoVidA - Pen-Based Collaborative Video Annotation

As an application for collaborative work, we have developed a pen-based annotation tool for videos. Annotating videos is an exhausting task, but 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 and labeling the shape of an object, we make use of the *Touch & Write* SDK.

5.1 Interaction

First, we introduce the touch-based gestures for the interaction with the video. The video media can be controlled via touch on the *Video Controls* and a drag gesture on the *Slider* (see below the video in Figure 5 where the 42 % is indicating the current slider position). The *Video Controls* provide the main functionality to navigate in the active video (fast backward or forward, play, pause, or stop). To navigate fast through the video the user can also use the slider by tapping to the desired position or fast searching via dragging.

Videos, annotation list and annotation fields can be manipulated via touch gestures. The standard functionalities are resizing, moving, or rotation (these gestures are similar to the well-known gestures on small Touch displays). These touch gestures provide a simple access for end-users. These manipulation operations are important as we are dealing with an environment for multiple users. As several users can be located around the tabletop, they need the functionality to adjust the videos to their needs. The touch-based gestures are detected by the *Touch & Write* server and triggered in the visual component via High-Level-Events. The *CoVidA* application receives these events and processes the event data to manipulate the videos in excepted ways.

5.2 Annotations

Annotations were performed by using the Anoto pen. Similar to a real pen, the anoto pen is used for outlining the desired shapes and for writing the annotation terms. Here we differentiate between a *frame annotation* and an *object annotation*.

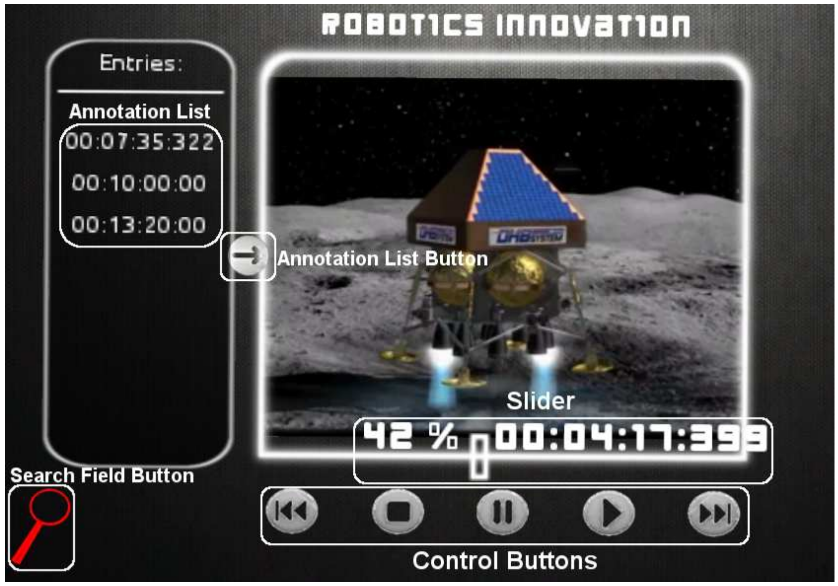

Fig. 5. Touch-Based Interactions: Users can interact with the buttons and annotation entries with simple single touches.

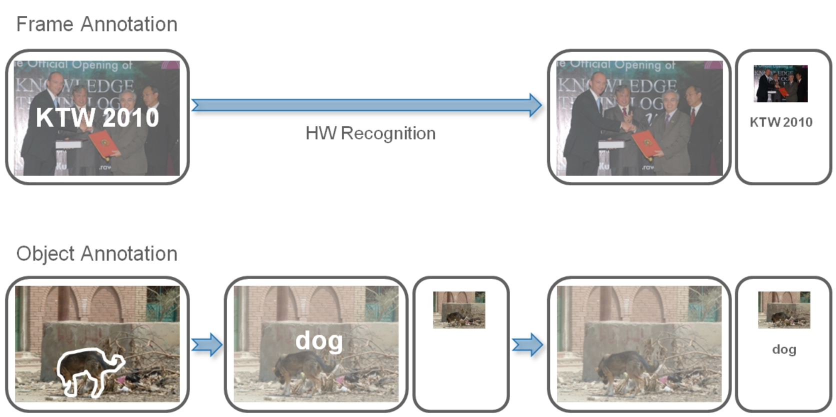

Fig. 6. Video Annotation Process

A *frame annotation* is associated with the whole frame and an *object annotation* is associated to a specific object which is outlined by a drawn shape. The user can annotate whole frames fast and simple by writing on the video using the pen device. The online mode detection classifies the pen input as text and the handwriting recognition is triggered. *CoVidA* now receives the recognition results as a ranked word list of the recognition results. The received top result is stored as description for the whole frame after touching on the save button (see Figure 6).

If the user starts with 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 an object annotation (see Figure 6, Bottom and By having this kind of automatism, users are not forced to switch the current mode manually.

5.3 Evaluation

The *CoVidA* system has been evaluated in a user study. The evaluation had three main goals:

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

The *CoVidA* software was evaluated against the *LableMe* application which is a well-recognized application by the MIT.

In our experiments we show that especially for complex structures the usage of a pen device improves the effectiveness of the outlining process. Furthermore, the toolkit was very intuitive and most users, even experts favored *CoVidA* over *LableMe*[8].

6 Conclusions

Touch & Write is a robust means for multi-touch and pen input providing the basis for a whole bunch of experimental research. It allows an intuitive switch between the modes object manipulation, and content editing. Online streaming provides direct feedback from the various applications. In this paper we have presented three examples where touching and writing are successfully combined in different application areas.

References

1. PyMT project homepage. (2010), http://pymt.txzone.net/
2. Ahmed, S., Weber, M., Liwicki, M., Dengel, A.: Improved automatic analysis of architectural floor plans. In: Proc. 10th ICDAR 2009. To appear in September 2009
3. Ahmed, S., Weber, M., Liwicki, M., Dengel, A.: Text/graphics segmentation in architectural floor plans. In: Proc. 10th ICDAR 2009. To appear in September 2009
4. Brooke, J.: SUS: A quick and dirty usability scale. In: Jordan, P.W., Weerdmeester, B., Thomas, A., Mclelland, I.L. (eds.) Usability evaluation in industry. Taylor and Francis, London (1996)

[8] This has been found by applying the System Usability Scale (SUS) [4]

5. Dietz, P., Leigh, D.: Diamondtouch: a multi-user touch technology. In: UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology. pp. 219–226. ACM, New York, NY, USA (2001)
6. Echtler, F., Klinker, G.: A multitouch software architecture. In: Proceedings of NordiCHI 2008 (Oct 2008)
7. Frisch, M., Heydekorn, J., Dachselt, R.: Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. In: ITS '09: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces. pp. 149–156. ACM, New York, NY, USA (2009)
8. Han, J.Y.: 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. pp. 115–118. ACM, New York, NY, USA (2005)
9. Kaltenbrunner, M., Bovermann, T., Bencina, R., Costanza, E.: Tuio - a protocol for table based tangible user interfaces. In: Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005). Vannes, France (2005)
10. Kara, L.B., Stahovich, T.F.: Sim-u-sketch: a sketch-based interface for simulink. In: AVI. pp. 354–357 (2004)
11. Laufs, U., Ruff, C.: MT4J project homepage. (2010), http://www.mt4j.org/
12. Leitner, J., Powell, J., Brandl, P., Seifried, T., Haller, M., Dorray, B., To, P.: Flux: a tilting multi-touch and pen based surface. In: CHI EA 09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems. p. 32113216. ACM, New York, NY, USA (2009), http://mi-lab.org/wp-content/blogs.dir/1/files/publications/int125-leitner.pdf
13. Liwicki, M., El-Neklawy, S., Dengel, A.: Touch & Write - A Multi-Touch Table with Pen-Input. In: Proceedings International Workshop on Document Analysis Systems (2010), to appear
14. Os, E.D., Boves, L.: Towards ambient intelligence: Multimodal computers that understand our intentions. In: Proc. eChallenges. pp. 22–24 (2003)
15. Spassova, L., Schöning, J., Kahl, G., Krüger, A.: Innovative Retail Laboratory. In: Roots for the Future of Ambient Intelligence. European Conference on Ambient Intelligence (AmI-09). o.A., Salzburg, Austria (November 2009)
16. Weber, M., Liwicki, M., Dengel, A.: a.SCAtch - A Sketch-Based Retrieval for Architectural Floor Plans. In: 12th International Conference on Frontiers of Handwriting Recognition. (2010)