TOUCHify: Bringing Pen-Based Touch Screen Functionality to Flat Panel Display Screens

Ihab Al Kabary Marcel Büchler Heiko Schuldt
Databases and Information Systems Group
Department of Mathematics and Computer Science
University of Basel, Switzerland
Email: firstname.lastname@unibas.ch

***Abstract*—The past years have seen a rapid proliferation of flat panel display (FPD) screens that are available in almost all households and that have almost completely displaced cathode ray tube screens. While these screens have a great potential for serving as unobtrusive user interaction devices, usually only a small portion of FPDs are built to incorporate touch screen functionality due to the cost it entails. Several attempts have been made to economically transform existing FPDs into touch-enabled screens using either fingers, pens or styluses. In this paper we present *TOUCHify*, a novel approach that uses a dynamic pixel pattern seamlessly integrated into an application's output on display, together with a digital microscopic pen. The pen is able to detect the embedded pixel pattern which, in turn, allows the application to react to user interaction. This enables pen-based touch screen functionality on arbitrary FPDs. An evaluation of the *TOUCHify* prototype with a healthcare feedback form emulator in an ambient assisted living scenario shows the effectiveness of the approach.**

***Index Terms*—human-computer interaction, pen-based user interaction**

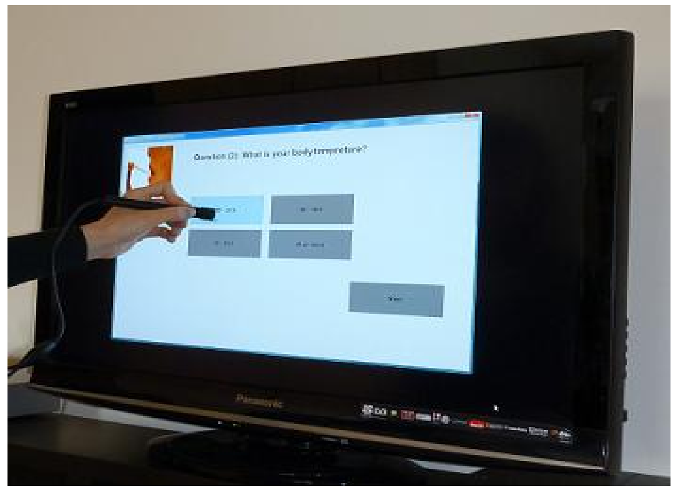


Figure 1. TOUCHify in action

I. Introduction

Flat panel display (FPD) screens, whether in TVs or PCs, have been very widely displacing cathode ray tube screens as the latter are no longer state of the art. This can be attributed to the less space FPDs require, less radiation they emit, less energy they consume, less flickering they generate and the better image quality they produce. Furthermore, a very small portion of these FPDs even have touch-based input capabilities (fingers, stylus or pen-based) like in surfaces or tablet PCs. However, the vast majority of these FPDs do not incorporate such functionalities mainly due to the cost factor. However, such functionality would be highly desired, especially for providing well accepted and broadly available devices for unobtrusive human computer interaction for rather technophobe users (e.g., in ambient assisted living environments, in eLearning applications, etc.). Hence, this has led to several attempts to economically transform existing widespread FPDs to act as touch-based input devices, besides their traditional function as an output device. These approaches rely either on tracking pens emitting infra-red light, measuring ultrasonic wave reflections off pens or reading patterns printed on a special film attached on top of the screen.

In this paper, we present *TOUCHify*, a novel approach that uses a digital microscopic pen and that leverages a dynamic pixel pattern that is seamlessly integrated into an application's output on display. The pen is able to detect the embedded pixel pattern which, in turn, is considered as input of the application on display and which allows the application to react to this user interaction. Consequently, this interaction enables pen-based touch screen functionality on arbitrary FPDs. We present the underlying concepts of *TOUCHify* and our implementation. Moreover, we have deployed *TOUCHify* in an ambient assisted living scenario by allowing user interaction on the basis of healthcare feedback forms. We present the results of user studies that show the effectiveness of the approach.

The remainder of this paper is structures as follows: Section II summarizes related work. In Section III, we discuss the concepts and implementation of *TOUCHify*. Results of the user study evaluation are presented in Section IV. Section V concludes.

II. Related work

Following the early vision of pen-based computing [18], various methods have been developed to enable FPDs (and in some cases even just any flat surface) to have touch functionality. The commercial and widely popular DUO pen [2] is a device that has the ability to convert any ordinary laptop into a tablet computer with full pen-based touch screen capability. It is composed of a base station that is placed on top of the screen

978-1-908320-13/1/$25.00©2013 IEEE

and that is equipped with ultra sound receivers to acquire the spatial coordinates of the pen. The pen has been reported to provide noticeable accuracy except for the top corners and needs initial calibration efforts. Mimio pens [7] work in a similar way but are rather used on larger surfaces to create interactive whiteboards.

Wii Remote [14] is a multi-point interactive whiteboard that has gained viral popularity, especially in the educational field. It uses the Nintendo Wii game console as a tracker of infrared (IR) light sources. By pointing a Wiimote at a projection screen or a FPD, the Wiimote can track up to four IR pens and thus create very low-cost, interactive whiteboards or tablet displays. Its main disadvantage is shadow casting, as standing between the Wiimote and the projection screen or FPD renders the system functionless. Another system to be launched soon is LEAP motion [5] which also relies on IR light sources and targets the smallers FPD screens.

Another approach mimics the Anoto [1] patterns on FPDs. Anoto is a proprietary and irregular dot pattern which consists of very small dots arranged on a grid with a spacing of approximately 0.3mm. It is used mainly to create interactive paper, where the user draws on paper using a pen [6] which is equipped with an infrared LED camera (100 fps) that can localize the position on paper by reading a 6×6 dot area (corresponding to an area of 1.8×1.8mm in size). It can also be used with projection screens and table-tops to create interactive areas and can even be printed on transparent foils and attached to FPDs as shown in DigiSketch [16]. Its main disadvantage is that the special film clutters the FPD screen to some degree giving it a slightly dimmed look. On the other hand, applications making use of the anoto pattern have found significant success in the interactive paper-based environment as shown in PaperPoint [20] and PaperProof [22].

There are plenty of other devices that enable pen or stylus-based data-input. Graphic tablets like the Wacom tablets [13] are widely used. However, their main disadvantage is the need for constant eye-hand coordination, which involves constantly looking away from the hand towards the screen, then re-aligning the hand to the intended position every time the hand is raised from the tablet. The Pegasus pen [8] mimics what is written or sketched on a paper to the computer. It uses technology similar to the DUO pen. The MEMO-PEN [19] is one of the pioneer approaches in pen tracking. It is an ordinary ball point pen except for a capability of memorizing what it draws in itself. The pen is equipped with a CCD camera and captures a series of partial snapshots in its memory and reconstructs them into a full image later. In this way it provides an asynchronous approach for data input. PenLight [21] combines a mobile projector and a digital pen to visually augment paper documents, giving the user immediate access to additional information and computational tools and relieving the user from looking at the screen for extended feedback or functionality. In the coming section we will discuss *TOUCHify*, our new approach for enabling pen-based touch screen functionality on any FPD.

Figure 2. Microscopic Pens next to a Standard Pencil

III. TOUCHify

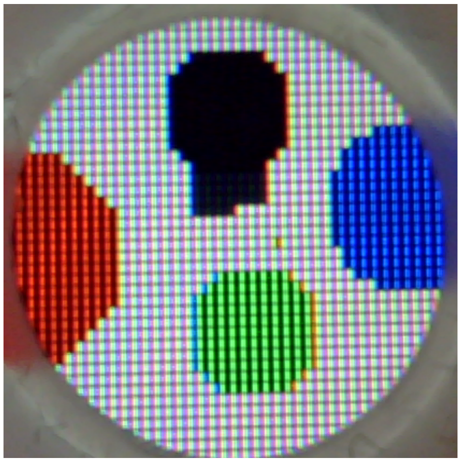
The *TOUCHify* prototype exploits a robust pixel pattern, dynamically embedded into the application's output to be displayed on the FPD. This pixel pattern allows to acquire location information from the screen so that a camera-equipped pen is converted into an input device. The following two subsections will explain the hardware and the software components of *TOUCHify*.

A. The Pen

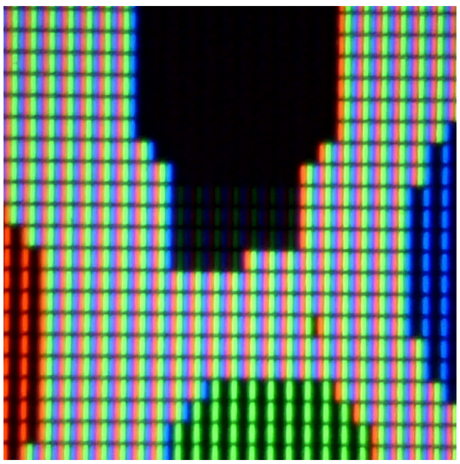
The camera for reading the pixel patterns that will be displayed on the FPDs has to be both accurate, with a reasonable frame rate and zooming capabilities and at the same time inexpensive. Digital microscopic pens as seen in Fig. 2 are the preferred devices over, for instance, web cams for two reasons. Firstly and most importantly, the latter usually have a rather high minimal focal distance that prevents a use as input device directly pointing to a screen. Secondly, the form factor of web cams usually prevents their use as input devices held by humans. Besides having the traditional, easy to handle pen design, they provide rather high zoom capabilities (up to 150 times), reasonable frame rates (up to 30 fps) and a rather low minimal focal distance of 1 cm. Communication is currently achieved via USB connection but Bluetooth versions are already in development.

B. The Pixel Pattern

The main objective of the pixel pattern, added dynamically and as indiscernible as possible to the output, is to identify location / GUI controls on the screen to enable pen-based touch functionality on FPDs. LCD screens are more commonly used as computer screens while plasma screens are more widely used in TVs. Both types of screens rely on the concept of subpixels to render images. Each pixel is usually composed of three individual red, green, and blue (RGB) subpixels to anti-alias text with greater detail and to increase the rendering quality. Recent commercial developments have introduced LCD screens Quattron [11] that utilize yellow as a fourth color subpixel (RGBY) claiming it increases the range of



(a) Pixels View (Lower Zoom)



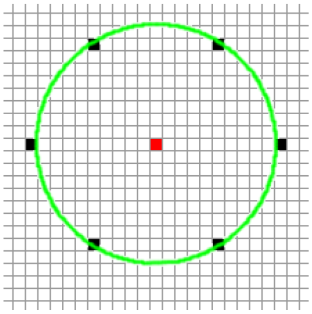
(b) Subpixels View (Higher Zoom)

Figure 3. Screenshots at Various Zoom Levels

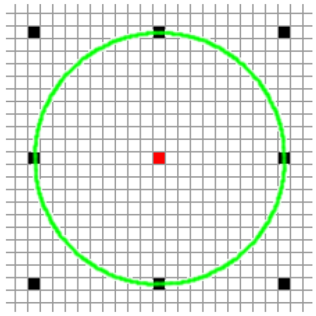
displayable colors in a way that mimics the way the brain processes color information. Similarly, PenTile [9] exploits the fact that the human eye is most sensitive to green. The RGBG scheme creates a color display with one third fewer subpixels than a traditional RGB-RGB thus saves more energy. When seen from a distance by the human eye, these subpixels blend to form pixels of a spectrum of colors. Due to the different subpixel compositions, we have decided to avoid using subpixels for rendering the pixel pattern. Instead, we have decreased the magnitude of the zoom of the microscopic pen to avoid working with subpixels and rather use the human-perceived pixel colors. Fig. 3 shows the subpixels seen at a high zoom level and the formulation of pixels at a much lower zoom level.

One of the main challenges when developing *TOUCHify* was to come up with a pixel pattern that does not clutter much of the display screen. A pattern similar to the one used by Anoto for interactive paper, for instance, cannot be replicated on FPDs due to its very fine granularity and density of 600 dpi. On display screens, which usually only have around 100 ppi, less resolution is present to imprint the pattern without increasing visual clutter. DigiSketch [16] does indeed use Anoto patterns but by printing it on special films covering the FPDs. The main disadvantage is that this approach clutters the screen to some degree giving an overall grayish color to the surface of the FPD and that the patterns are always visible even if the user does not want to initiate a touch interaction and only wants to view the display screen.

1) Pixel Pattern Design: In designing the pixel pattern, we identified two main situations where the pattern is needed to enable pen-based touch screen functionality: initially, when attempting to interact with GUI controls like push buttons, check boxes, etc.; and secondly, when attempting to write, draw or sketch. We refer to these two patterns as *control-detection pattern* and *drawing pattern*, respectively. Inspired from 2D barcodes (QR codes [10]), we developed both a triangular and a square pattern for the control-detection pattern and the drawing pattern. In this paper we focus on the control-detection pattern which uses a triangular pattern encoded with color information to distinguish the different controls on screen. To establish this pattern, we initially capture the video frames from the microscopic pen at 30 fps. For accessing the camera and capturing the images, we used either JMF [3] or v4l4j [12] and both performed well. Each video frame captures the pixels on screen and each screen pixel can be seen as a patch of image pixels. The size of the patch will be dependent on the zoom-level and screen-resolution and thus needs to be calibrated initially. We call the patch of image pixels that represent a pixel on screen a *logical pixel*. For every video frame captured, we initially apply a Gaussian blur filter [4] with a rather small radius to slightly smoothen the image. The radius is 1.5 times the size of the logical pixel and has been chosen after empirical observations. This step is required to decrease the detection of false positive pattern pixels by decreasing the effect of the grid-like structure that appears between the individual pixels. We then apply our simplified implementation of the Canny edge detector [15] to assist in extracting the candidate pattern pixels. The simplified version scans the image in only one direction, opposed to the traditional implementation of the Canny edge detector that scans the image in four directions to detect edge points, thus, significantly increasing the speed of the detector. After detecting the edge pixels, several heuristics are used to identify the candidate pixel patterns. Most importantly, is to merge very closely clustered edge pixels which is needed because of the one directional canny edge detector then to check that the merged-pixels have the adequate size of the logical pixels. Finally, an array of these candidate pattern pixels are inspected to see if they form either a *control-detection pattern* or a *drawing pattern*. A control-detection pattern has a pattern pixel surrounded by 6 pattern pixels at the specific pattern radius, separated by 60 degrees angles apart. On the other hand, a drawing pattern has 4 pattern pixels surrounding the pixel at the center of the image and have these pixel patterns separated by 90 degrees angles as shown in Fig. 4. In this way, both patterns are rotation invariant. If a control-detection pixel pattern is recognized, color information that has been initially



(a) Triangular Pixel Pattern



(b) Squared Pixel Pattern

Figure 4. A fast method to distinguish between the triangular and squared pixel patterns used for the control-detection and drawing patterns respectively

mapped to each control automatically by the PatternManager class is used to figure out which control has been interacted with.

2) Embedding TOUCHify: It is very straighforward to embed and enable *TOUCHify* in a GUI, e.g., by using the standard Java GUI classes when developing an application in Java. There are no restrictions on how the GUI is designed. Any Java layout manager can be used. For the controls, the PatternButton class is used instead of the traditional button. It is merely a subclass of the JToggleButton class and it simply just overlays the standard button with the *TOUCHify* pixel pattern. The only specific classes that need to be used to enable *TOUCHify* are the PatternImageProcessor and the PatternManager. The PatternImageProcessor is responsible for capturing the video frames and detecting the pattern pixels. The PatternManager object is linked to various GUI controls.

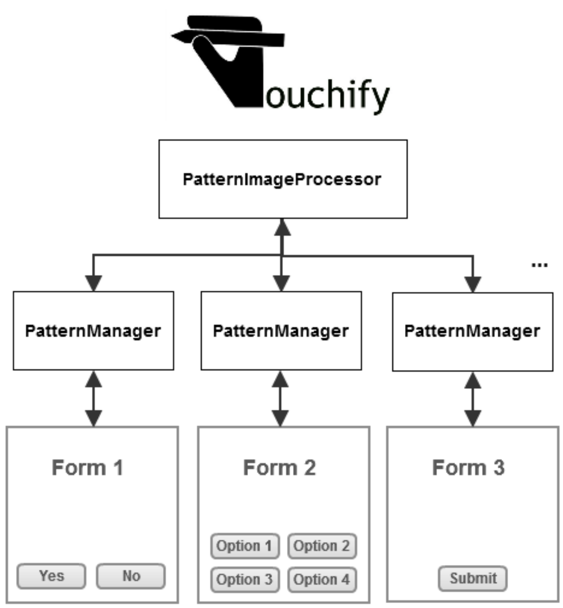


Figure 5. TOUCHify Class Interaction Diagram

It takes the pixel patterns detected from the PatternImageProcessor and maps it to the correct control on screen and triggers the click event. In this way, each application can have a single PatternImageProcessor object and potentially several PatternManager objects for each window frame with its set of GUI controls as depicted in Fig. 5.

IV. Evaluation

Pen-based functionality can be very useful in a variety of use cases. In order to evaluate the effectiveness of the system, we have chosen an ambient assisted living scenario.

A. Healthcare Use Case

Elderly, rather technophobe users are supposed to fill in simple questionnaires assessing their health state on a regular basis (e.g., one to three times a day) and this information is usually transmitted automatically to their health care provider. In order to provide this information, devices should be embedded as unobtrusively as possible into their home environment.

In the *TOUCHify* approach, they only need their home TV set and the *TOUCHify* pen. The main advantage of *TOUCHify* in this scenario is that users can use the devices they are very familiar with, which are already present in their daily environment and which anyway attract their attention throughout the day. To accomplish this, we have developed a Healthcare feedback form emulator that provides a way to easily build questionnaires that are *TOUCHify* compliant. We have designed a questionnaire comprising of a start-up form that gains attention of the patient visually and optionally acoustically through a sound clip. As soon as the patient clicks the button using the *TOUCHify* pen, 5 questions in multiple choice (MCQ) or true/false format appear in succession. In each of these forms, a next button is activated as soon as the user chooses any option, giving the user time to change the chosen option if needed. At the end of the 5 questions, a review form is presented with all the questions and the chosen answers providing a final chance to review the answers before potentially sending them to the healthcare provider via the Internet. Screenshots for samples of the questionnaire forms can be viewed in Fig. 6.

B. Evaluation Results

In the evaluation, we asked 11 users to participate in answering the questionnaire. No training was given to the users, just an introduction about the *TOUCHify* project and simple instructions about their role in answering the questionnaire. It was stated to each one of them that it did not matter what answer they gave to the questions but rather how precise the *TOUCHify* system worked. It was also stated that the pen needs to be held in a near orthogonal direction to the screen without too much tilting. As soon as the user starts the questionnaire, a timer is activated that helps us track how much time was needed for the user to complete the questionnaire. This information was not given to the user to avoid any pressure in trying to finishing the evaluation session rapidly. During the evaluation session, we manually count the

TABLE I
EVALUATION SESSION RESULTS FOR THE UNTRAINED TESTING USERS

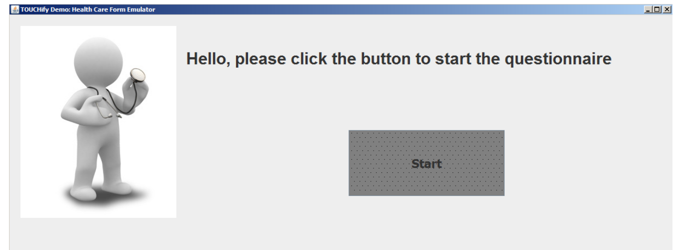
User No.	Retry clicks	Ellapsed time	Ease of use
1	2	58.64 s	2
2	3	38.92 s	2
3	3	81.47 s	2
4	4	52.67 s	2
5	0	26.04 s	1
6	0	35.02 s	1
7	0	55.33 s	2
8	4	76.94 s	3
9	3	50.46 s	4
10	1	48.74 s	3
11	3	76.03 s	2

number of clicks where the user had to retry clicking the button to trigger a button click event. At the end of the evaluation sessions, each user was asked to rate the system on a scale from 1 to 5 with 1 being very easy and 5 being very hard. Then the users were free to give us feedback on their experience with *TOUCHify*. A summary of the evaluation sessions are shown in Table I and Table II.

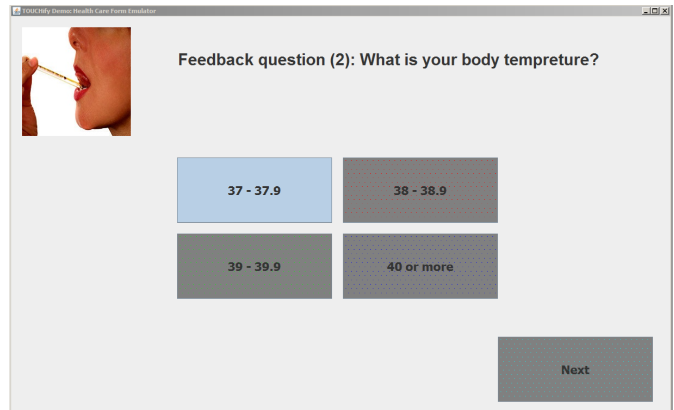
All the 11 users managed to complete the questionnaire and the results of our prototype were very promising and have shown the effectiveness of the approach. By inspecting the data in Table I, we can judge that the system was rather easy to

TABLE II
FEEDBACK ISSUES AND NUMBER OF USERS MENTIONING IT

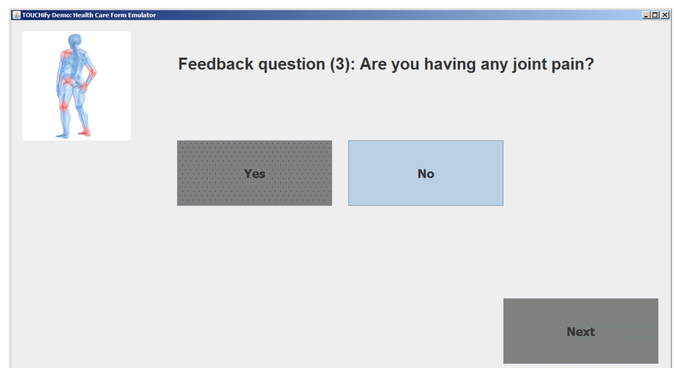
Feedback topic	No. of users
The need for the near orthogonal hold of the pen against the screen is inconvenient	9
The need for a wireless pen	4
The need for a more ergonomic design for the pen	3
The evident appearance of the dot pattern on the buttons controls is inconvenient	2



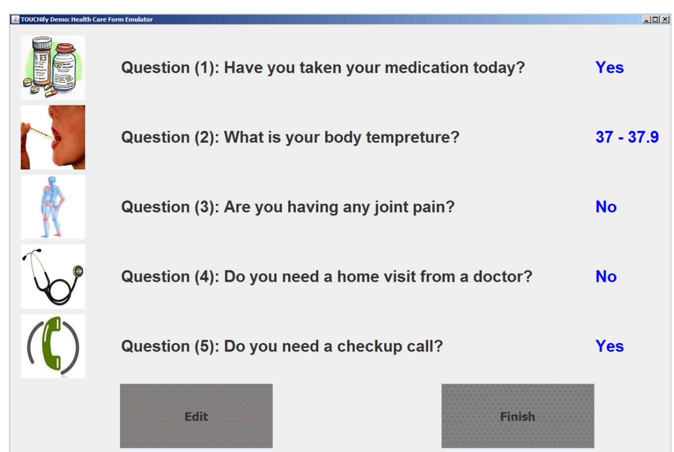
(a) Initial Wizard Form



(b) Sample (Yes / No) Question



(c) Sample Multiple Choice Question (MCQ)



(d) Sample Multiple Choice Question (MCQ)

Figure 6. Healthcare State Feedback Forms

use but should be enhanced to be even easier. The users took less than a minute on average to answer the questionnaire, which is very reasonable as they read through the questions despite the fact that we explicitly told them that any answer will be a good answer. Several users experienced undetected clicks and had to try clicking again and were successful the second try. From our observation, we noticed that this was mainly due to two factors: either a tilted pen position or when the pixel pattern color was the same as the foreground text on the buttons.

Regarding the feedback we got from the users after completing the questionnaire, and focusing only on the negative comments, the three most inconvenient aspects of the system were the need for a near orthogonal hold of the pen to the screen, the need for a wireless pen and the need for a more ergonomic design of the pen. For the first issue, we can attempt to apply image processing techniques to correct the image deformation resulting from the various tilting positions. For the wireless pen issue, we are awaiting the release of the Bluetooth/WiFi microscopic pens and hopefully higher frames per second for more accuracy. A few users also wanted a better ergonomic design for the pen. They wanted it to have a rather thin tip, so they can accurately pinpoint what they are selecting. Furthermore, relating to the ergonomic design, having a rubber coating on the tip of the pen was also highlighted by one of the test users as he feared that if the pen is excessively used it could damage the screen. Finally, a couple of users found the appearance of the patterns to be rather evident and inconvenient. The feedback we got from the various users gave us a much clearer understanding on what to focus on in order to further develop *TOUCHify* and make it more efficient and user friendly. The latter issue, the visibility of the pattern, will be addressed in future work by applying steganographic techniques to hide the pixel pattern as much as possible and even avoid showing the pixel patterns until the pen is close enough to the screen.

V. Conclusions and Outlook

TOUCHify allows to significantly enrich passive FPDs by adding pen-based touch screen functionality. In a first version, it has been used together with a control-detection pattern for simplifying form-based user interaction. With the more advanced drawing patterns, novel types of applications on FPDs will be supported, like sketch-based human-computer interaction (e.g., for more customized healthcare applications, or for completely different fields such as sketch-based multimedia retrieval [17]). The system has been put to the test and has shown its effectiveness by using it as the input device for a healthcare feedback application in which patients, especially the elderly, can use their own TV sets to answer questionnaires regarding their health and potentially have this information relayed back to the clinic via the Internet.

In future work, we plan to extend *TOUCHify* to make it even more user friendly by enabling it to work under very tilted pen positions. Using the CIELAB color model instead of RGB in picking colors for the pixel pattern with small deltas to the background that humans have difficulty distinguishing could also be beneficial in hiding the pattern. A wireless pen, designed in a more ergonomic manner especially for this task, could also make *TOUCHify* even more user friendly. *TOUCHify* can also benefit from automatic calibration that could tweak parameters like logical pixel size. This is needed since each FPD can have various screen resolutions and setting this manually requires empirical observations. Another interesting idea is to further investigate how we can extract location information from sub-pixels. This could further assist us in hiding the pixel patterns and eliminate the need for various steps needed in detecting the pattern like blurring and detecting logical pixels. Currently, a click event requires the user to float on the button for a short time; alternatively, the system has to wait until the user moves away from the button. We plan to customize the pen hardware to have a pressure sensitive tip in order to facilitate click events in *TOUCHify*.

Acknowledgments

This work has been partly supported by the Swiss National Science Foundation, projects PAD-IR and MM-DocTable.

References

[1] Anoto. http://www.anoto.com. Accessed: 2013-04-29.

[2] DUO pen. http://www.penandfree.com. Accessed: 2013-04-29.

[3] Java Media Framework. http://java.sun.com/products/java-media/jmf. Accessed: 2013-04-29.

[4] JHLabs. http://www.jhlabs.com/ip/filters. Accessed: 2013-04-29.

[5] LEAP motion. http://www.leapmotion.com. Accessed: 2013-04-29.

[6] Livescribe Pens. http://www.livescribe.com. Accessed: 2013-04-29.

[7] Mimeo pens. http://www.mimio.com. Accessed: 2013-04-29.

[8] Pegasus pens. http://www.pegatech.com. Accessed: 2013-04-29.

[9] PenTile. http://www.nouvoyance.com/technology.html. Accessed: 2013-04-29.

[10] QR Code. http://www.qrcode.com. Accessed: 2013-04-29.

[11] Quattron. http://www.quattron.tv/technology. Accessed: 2013-04-29.

[12] v4l4j Library. http://code.google.com/p/v4l4j. Accessed: 2013-04-29.

[13] Wacom tablets. http://www.wacom.com. Accessed: 2013-04-29.

[14] Wiimote. http://www.johnnylee.net/projects/wii/. Accessed: 2013-04-29.

[15] J. Canny. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, June 1986.

[16] R. L. Hofer and A. Kunz. Digisketch: Taming Anoto Technology on LCDs. In *Proc. of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS'10)*, pages 103–108. ACM, 2010.

[17] R. Kreuzer, M. Springmann, I. Al Kabary, and H. Schuldt. An Interactive Paper and Digital Pen Interface for Query-by-Sketch Image Retrieval. In *Proceedings of the 34th European Conference on Information Retrieval (ECIR'12)*, volume 7224 of *LNCS*, pages 317–328. Springer, Apr. 2012.

[18] A. Meyer. Pen Computing: a Technology Overview and a Vision. *SIGCHI Bulletin*, 27(3):46–90, July 1995.

[19] S. Nabeshima, S. Yamamoto, K. Agusa, and T. Taguchi. Memo-pen: a new input device. In *Conference companion on Human factors in computing systems*, CHI '95, pages 256–257, NY, USA, 1995. ACM.

[20] B. Signer and M. C. Norrie. PaperPoint: a Paper-based Presentation and Interactive Paper Prototyping Tool. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction 2007*, pages 57–64, , Baton Rouge, Louisiana, USA, 2007.

[21] H. Song, T. Grossman, G. W. Fitzmaurice, F. Guimbretiere, A. Khan, R. Attar, and G. Kurtenbach. PenLight: Combining a Mobile Projector and a Digital Pen for Dynamic Visual Overlay. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI 2009)*, pages 143–152, Boston, MA, USA, Apr. 2009. ACM.

[22] N. Weibel, A. Ispas, B. Signer, and M. C. Norrie. Paperproof: a Paper-digital Proof-editing System. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems (CHI 2008)*, pages 2349–2354, Florence, Italy, Apr. 2008.