P

Pen–Based Digital Screen Interaction

Khaireel A. Mohamed
Albert-Ludwigs-Universität Freiburg, Germany

Thomas Ottman
Albert-Ludwigs-Universität Freiburg, Germany

INTRODUCTION

Through a transducer device and the movements effected from a digital pen, we have a pen-based interface that captures digital ink. This information can be relayed on to domain-specific application software that interpret the pen input as appropriate computer actions or archive them as ink documents, notes, or messages for later retrieval and exchanges through telecommunications means.

Pen-based interfaces have rapidly advanced since the commercial popularity of personal digital assistants (PDAs) not only because they are conveniently portable, but more so for their easy-to-use freehand input modal that appeals to a wide range of users. Research efforts aimed at the latter reason led to modern products such as the personal tablet PCs (personal computers; Microsoft Corporation, 2003), corporate wall-sized interactive boards (SMART Technologies, 2003), and the communal tabletop displays (Shen, Everitt, & Ryall, 2003).

Classical interaction methodologies adopted for the desktop, which essentially utilize the conventional pull-down menu systems by means of a keyboard and a mouse, may no longer seem appropriate; screens are getting bigger, the interactivity dimension is increasing, and users tend to insist on a one-to-one relation with the hardware whenever the pen is used (Anderson, Anderson, Simon, Wolfman, VanDeGrift, & Yasuhara, 2004; Chong & Sakauchi, 2000). So, instead of combining the keyboard, mouse, and pen inputs to conform to the classical interaction methodologies for these modern products, our ultimate goal is then to do away with the conventional GUIs (graphical user interfaces) and concentrate on perceptual starting points in the design space for pen-based user interfaces (Turk & Robertson, 2000).

BACKGROUND

If we attempt to recognize the digital pen as the only sole input modal for digital screens, for both interfacing and archival modes purported within the same writing domain, we then require the conceptualization of a true perceptual user interface (PUI) model. Turk and Robertson (2000) discuss the main idea of having an alternative (graphical user) interface through the PUI paradigm as a nonhassled and natural way of communicating with the background operating system. It is subjective, and it concerns finding out and (to a certain extent) anticipating what users expect from their application environment. There are several reasons to utilize the PUI as an interactive model for the digital screen. Amongst some of the more prominent ones are the following:

- To reintroduce the natural concept of communication between users and their devices
- To present an intelligent interface that is able to react accordingly (as dictated by the objective of the application program) to any input ink strokes
- To redesign the GUI exclusively for perceptual conceptions

Modern and networked interactive digital screens utilize the electronic pen's digital ink as a convenient way of interfacing with specially developed application programs, and go on to offer the visual communication of opinions for multiple users. This is as a result of taking advantage of the pen-based environment. For example, we want to reproduce the simple, customary blackboard and still be able to include all other functionalities that an e-board can offer. But by minimizing the number of static menus and buttons (to accommodate new perceptual designs in accordance to the PUI standards), the resultant "clean slate" becomes the only perceptual input available to users to relate to the background systems. Here, we see two distinct domains merged into one: the domain to receive handwritings (or drawings) as the symbolic representation of information (termed technically as traces), and the domain to react to user commands issued through pull-down menus and command buttons.

Based purely on the input ink traces, we must be able to decipher users' intentions in order to correctly classify which of the two domains it is likely to be in: either as primitive symbolic traces, or some sort of system command. Often, these two domains overlap and pose the problem of ambiguousness, a gray area that cannot be simply classified by means of straightforward algorithms. For instance, the background system may interpret a circle drawn in a clockwise direction over some preexisting ink traces as a select command when in fact the user had simply intended to leave the circle as a primitive ink trace to emphasize the importance of his or her previously written points. Fortunately, this problem can be solved if the program can anticipate the intentions of its users (Wooldridge, 2002); however, this method necessitates the constant tracking of the perceptual environment and would require a more stringent and somewhat parallel structural construct in order to run efficiently (Mohamed, 2004b; Mohamed, Belenkaia, & Ottman, 2004).

There are currently many works by authors that describe vividly the interpretations of these traces exclusively in either domain as well as in combination of the two. In the trace-only domain, Aref, Barbara, and Lopresti (1996) and Lopresti, Tomkins, and Zhou's (1996) collective research in dealing with a concentrated area of deciphering digital inks as hand-drawn sketches and handwritings, and then performing pictorial queries on them, is the result of their effective categorization of ink as a "first-class" data type in multimedia databases. Others like Bargeron and Moscovich (2003) and Götze, Schlechtweg, and Strothotte (2002) analyze users' rough annotations and open-ended ink markings on formal documents and then provide methods for resetting these traces in a more orderly, cross-referenced manner. On the opposite perspective, we see pilot works on pen gestures, which began even before the introduction of styluses for digital screens. They are purported on ideas of generating system commands from an input sequence of predetermined mouse moves (Rubine, 1991). Moyle and Cockburn (2003) built simple gestures for the conventional mouse to browse Web pages quickly, as users would with the digital pen. As gesturing with the pen gained increasing popularity over the years, Long, Landay, Rowe, and Michiels (2000) described an exhaustive computational model for predicting the similarity of perceived gestures in order to create better and more comfortable user-based gesture designs.

For reasons of practicality and application suitability, but not necessarily for the simplicity of implementation, well-developed tool kits such as SATIN (Hong & Landay, 2000) and TEDDY (Igarashi, Matsuoka, & Tanaka, 1999) combine the pen input modality for two modes: sketching and gesturing. The automatic classification of ink inputs directed for either mode do not usually include too many gestures, and these tools normally place heavier cognition loads on the sketching mode. We agree that incorporating a pen-based command gesture recognition engine, as a further evaluation of the input traces and as an alternative to issuing system commands for addressing this scenario, is indeed one of the most practical ways to solve the new paradigm problem.

ISSUES ON REPRESENTING DIGITAL INK TRACES

A trace refers to a trail of digital ink data made between a successive pair of pen-down and pen-up events representing a sequence of contiguous ink points: the *X* and *Y* coordinates of the pen's position. Sometimes, we may find it advantageous to also include time stamps for each pair of the sampled coordinates if the sampling property of the transducer device is not constant. A sequence of traces accumulates to meaningful graphics, forming what we (humans) perceive as characters, words, drawings, or commands.

In its simplest form, we define a trace as a set of (x_i, y_i, t_i) tuples, deducing them directly from each complete pair of pen-down and pen-up events. Each trace must be considered unique and should be identifiable by its trace ID (identification). Figure 1 depicts the object-oriented relations a trace has with its predecessors, which can fundamentally be de-

Related Content

Cultural Property and Identity Issues in International Law: The Inadequate Protection of the Cultural Heritage of Indigenous Peoples
Athanasios Yupsanis (2012). *Human Rights and Risks in the Digital Era: Globalization and the Effects of Information Technologies (pp. 256-277).*
www.irma-international.org/chapter/cultural-property-identity-issues-international/64949

Playing Virtual Power Games: Micro-Political Processes in Inter-Organizational Networks
Monique Janneckand Henning Staar (2011). *International Journal of Social and Organizational Dynamics in IT (pp. 46-66).*
www.irma-international.org/article/playing-virtual-power-games/53825

'Wiring' African Newsrooms: The Internet and Mainstream Print Journalism Practice in Zimbabwe
Hayes Mawindi Mabweazara (2011). *Cultural Identity and New Communication Technologies: Political, Ethnic and Ideological Implications (pp. 144-162).*
www.irma-international.org/chapter/wiring-african-newsrooms/53771

Information Structure Parsing for Chinese Legal Texts: A Discourse Analysis Perspective
Bo Sun (2019). *International Journal of Technology and Human Interaction (pp. 46-64).*
www.irma-international.org/article/information-structure-parsing-for-chinese-legal-texts/214930

Investigating Smartphone Brand Loyalty for Millennials and Gen Z: A Customer Value Perspective
Masood H. Siddiquiand Tripti Ghosh Sharma (2022). *International Journal of Technology and Human Interaction (pp. 1-19).*
www.irma-international.org/article/investigating-smartphone-brand-loyalty-for-millennials-and-gen-z/302664