Proceedings of the 16th Annual NACCQ, Palmerston North New Zealand July, 2003 (eds) Mann, S. and Williamson, A. www.naccq.ac.nz

Do Tablets dream of electric ink?

Todd Cochrane
School of Information Technology
Wellington Institute of Technology
Petone, NZ
Todd.Cochrane@weltec.ac.nz

ABSTRACT

The Tablet PC version of Microsoft's Windows XP operating system includes components that recognise handwriting. The data structures and objects used to store handwriting in Tablet PCs are called "ink". The recognition of handwriting appears to be quite slow. A technical introduction to ink and the definition of the interfaces that allow the creation of purpose built handwriting recognition modules is described. These are then used to describe a proposed technical solution that speeds up handwriting recognition by embedding the recognition software in field programmable gate arrays (fpgas) mounted on a PCMCIA card. The solution then provides potential co-processing of handwriting recognition allowing recognition to proceed as a background process. Hence, daydreaming of electric ink becomes possible for the Tablet PC!

Keywords

Tablet PC, handwriting recognition, ink, new object class libraries, fpga, PCMCIAs, co-processor.

1. INTRODUCTION

A "Tablet PC" is computer hardware developed to a specification compatible with Microsoft's "Windows XP Tablet PC Edition" operating system. The Tablet PC revisits pen-based input, an established style of communication in digital media [1] [2]. It is a portable computer with: handwriting recognition, speech recognition and wireless networking. Multimodal input[11] of this type provides the computer user a number of opportunities to interact with digital media in a natural way [4]. Pen based input has been considered and working computer input developed since 1963; Ivan Sutherland's PhD thesis "Sketchpad, a man-machine graphical communication system." Describes SketchPAD as "a novel communication medium for a computer" in which the user " sketches directly on a computer display with a'light pen'"([5] page 2) . Commercially available tablet format computers having a display and a writing tablet integrated into one device with: pen based input and handwriting recognition in a portable computer, have been available since in 1989 [3]. A number of "Tablet PC" computers are currently produced and marketed by major computer manufacturers and retailers, see [1] for a comprehensive list.

Multimodal interaction with computing systems in the form of the "Tablet PC" has become more feasible due to advances in software and hardware. Advances in the software development, include, for example: improvements in skeletonizing algorithms for offline handwriting recognition [6]. Advances in hardware include: fast graphics processors and central processing unit instruction execution speed. While current technological improvements make multimodal interaction more feasible, utimately the general purpose nature of the processors present a limit on peformance. This bottle-neck applies to both central processing

units and co-processors developed for a specific domain – for example graphics processing devices. This paper proposes the development of purpose built co-processors for handling specific human-computer interaction requirements. Development of a co-processor for handwriting recognition on a "Tablet PC" is used as a particular example.

2. HANDWRITING RECOGNITION AND THE TABLET PC PLATFORM

Handwriting recognition can be undertaken "offline" or "online". "Offline" handwriting recognition deals with data scanned from written documents. Features of characters and words are extracted from rasterized, pixel based data(see [6]). "Online" handwriting recognition extracts text from data that describes the physical movement of a stylus as it is used to write (see [7]).

An online handwriting recognition framework, Microsoft Corporation's "Tablet PC Platform" , adds handwriting recognition to the Window's XP operating system. In this "platform": the "Pen API" captures motion, from stylus movement on a tablet, passing coordinates to the "Ink API", the "Ink API" renders and stores motion in a data structure called "ink", the "Ink API" groups "strokes" and passes them to a recogniser, the "Recognizer API" returns a data structure called the "lattice", representing candidate text interpretations of the "Ink" (see [9][10]).

In an application on the "Tablet PC", handwriting recognition is achieved by creating instances of objects in the "Tablet PC Platform". The "Tablet PC Platform" provides a layer through which all interaction with stylus input using tablets must proceed. Each application that makes use of handwiting recognition has to be "ink-enabled". The layer provides a list of tablet devices and a list of handwriting recognisers.

2.1 Collecting Ink

Each window in an "ink-enabled" application can accept input from a number of tablets, as listed in the "Tablets Collection". The motion data is collected by an instance of an "Ink Collector", attached to the window, that captures input and directs it to an "Ink Object".

2.2 Managing Ink Data

The "Ink Object" contains a "Strokes Collection". Each "Stroke" represents a single "pen down" , "pen move" and "pen up" sequence. The Stroke is represented by a Bezier curve. The Bezier curve can "smooth" some sharp changes in stokes, "cusps", for example, in the lettter "L", the corner of the letter "L" is considered a "cusp". Strokes are divided according to "cusps" by the underlying "Ink Collection" object prior to being stored in Bezier form.

Each "Stroke" is stored as an instance of a "Stroke Object", which contains a collection of packets. A packet is the set of data a tablet sends at each sample point sample, for example, coordinate and pressure information.

2.3 Recognising Ink

A collection of strokes is sent to a recognition engine that returns a recognition result object. Each recognition engine is registered with the "Tablet PC Platform". These are "dll" based applications written to the "Recognition API" specification. An "Ink Enabled" application selects the "recognition engine" it will use, for example the "Tablet PC Input Panel" will only use a recognition engine that has been approved by Microsoft Corporation.

The recognition engine is required to supply a number of handles that provides a series of hooks to functions required by the "Tablet PC Platform" (see [8][9]). The recogniser context can facilitate "synchronous" or "asynchronus" recognition. Asynchronous recognition is achieved by providing call-back hooks that are called by the recognition engine when it is ready.

3. CONNECTING TO THE HARDWARE

Field programmable gate arrays (fpga) technology provides reconfigurable logic based hardware. A small number of gates is used to make a logic circuit, for example a NOR gate. Current fpga devices contain millions of gates. Fpgas are packaged into reconfigurable units that include onboard memory, serial and parallel ports, a clock, and standard bus connections to correct to, for example, a USB or PCMCIA bus (see [12]). For this paper a PCMCIA mounted fpga that has algorithms for handwriting recognition embedded on it is communicated with through a purpose built recognition engine.

On the Windows platform software for communication with PCMCIA is created using the device driver development kit (DDK). Example code for communicating with the fpga device is provided by the manufacturer[12].

4. HANDWRITING PREPROCESSING AND CLASSIFIER

During online handwriting recognition a number of techniques are brought together to produce text. During preprocessing, sample point data is "cleaned", for example, sampled points for a stroke can be normalized by smoothing to Bezier curve then resampled. Following preprocessing, stroke data is used to identify features that then improve selection of characters. Sample points can be used to create parametric models using attributes such as: change in x and slope between previous and current sample point. Parametric models, known as hidden Markov models (HMM), of characters can developed making use of large samples of handwriting. During recognition, parameter features are extracted from normalized stroke samples, these are then fed into HMMs for characters and matching scores are returned.The scores are used to classify strokes.

Handwriting recognition is inherently sequential and has with large number of relatively simple data points that get manipulated in simple yet computationally intensive ways. The fpga can be configured to work in a pipeline with many operations executing at the same clock cycle.

5. EXPRESSING ALGORITHMS IN HARDWARE

Developing logic circuits that express HMM would be tedious. Fortunately, high level programming systems are available. " Handle C " is a C like programming language that has been extended to include parallel statement execution blocks. Each statement in a "par block" is executed in parallel. Hence. SWAP becomes:

```
a = b;
b=a;
```

A more common language for expressing hardware is VHDL, "VHSIC (Very High Speed Integrated Circuits) Hardware Description Language". VHDL describes: hardware structure, dataflow and behavior. Of particular interest are expressions on signal firing. Signal change statements do not execute sequentially they execute according to a timing scheme, for example, a clock cycle. All changes defined by a series of signal statements continue to take effect until until there are no more changes in dependent signals. VHDL sequential statements can be expressed in "process blocks", each process block can be made sensitive to a set of signals, only executing when a signal changes. Process blocks are one way pipeline or parallel execution can be expressed in VHDL.

6. SUMMARY

High level hardware description languages bring HMM and sample point smoothing in hardware within the grasp of the software developer. In hardware:

* the limit of execution time for the sequentially executed or even partially pipelined general purpose instruction set can be reduced by executing many instructions in parallel

* application specific instructions can be executed in one clock cycle producing more efficient computing.

A true digital medium implies and requires transparent ubiquitous interaction with computing devices. One way to meet such requirements is purpose built co-processors.

REFERENCES

[1] Microsoft Corporation, 2003 "Windows Xp Tablet PC Edition Home Page" www.microsoft.com/windowsxp/tabletpc/default.asp Accessed 12 May 2003.

[2] Bajarin T, 2001, "ABCNEWS.Com: Silicon in Sights: The Tablet PC Returns" abcnews.go.com/sections/business/DailyNews/SILICON_INSIGHTS_BAJARIN_011220.html Accessed 29 May 2003

[3] GRiD Defense Systems Limited "Virtual Museum" www.griduk.com/muesem.htm Accessed 25 May 2003

[4] Oviatt S, Cohen P, 2000, "Multimodal Interfaces That Process What Comes Naturally" Communications of the ACM 2000.

[5] Sutherland I E, 1963 "Sketchpad, a man-machine graphical communication system" PhD Thesis, Massachusetts Institute of Technology, January 1963.

[6] Senior, A W, 1994, "Off-line Cursive Handwriting Recogniton using Recurrent Neural Networks" PhD Thesis ,University of Cambridge, England.

[7] Connell S D, 2000, "Online Handwriting recognition using multiple pattern class models" PhD Dissertation , Department of Computer Science and Engineering, Michigan State Unhiversity, USA.

[8] Microsoft Corporation , 2003 "Creating a Recognizer" http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tpcsdk10/html/

appendix/ tbconcustom recognizer.asp ?frame=true Accessed 12 May 2003

[9] Microsoft Corporation, 2003 “Recognition API Architecture” http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tpcsdk10/html/appendix/tbconrecointerfaces.asp Accessed 12 May 2003.

[10] Micorsoft Corporation, 2003 “Tablet PC Platform” http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tpcsdk10/html/managed/tbidxreferencesopener.asp Accessed 12 May 2003

[11] Oviatt S, Cohen P, 2000 “Multimodal interfaces that process what comes naturally” Communication sof the ACM March 2000/Vol 43, no 3.

[12] Annapolis MicroSystems 2003, “Wildcard”, http://www.annapmicro.com/products.html Accessed May 12 2003