From handwriting analysis to pen-computer applications

*

Lambert Schomaker
(schomaker@nici.kun.nl)
NICI/Nijmegen University, The Netherlands
Tel: +31-24-3616029, Fax: +31-24-3616029

Abstract

In this paper, **pen computing**, i.e., the use of computers and applications in which the pen is the main input device, will be described from four different angles. In the first section, a brief overview will be given on the hardware developments in pen systems. After concluding that the technological developments in this area did not lead to the expected user acceptance of pen computing, the reasons underlying this market failure are explored: The second part deals with Pen-User Interfacing (PUI) aspects. Problems of pen-user interface design are described. Existing and new applications are summarized. The third part is concerned with the handwriting process and product. The last part deals with automatic recognition methodologies. Four basic factors determining handwriting variation and variability are identified. A handwriting recognition approach using segmentation into velocity-based strokes is handled in somewhat more detail. A large-scale project (UNIPEN) concerns the benchmarking of the performance of on-line handwriting recognition algorithms which is crucial for the advancement of the state of the art in this area.

1 Introduction

Handwriting recognition and pen computing are characterized by an arduous evolution history. Originally identified thirty years ago as a first step towards the more difficult problem of speech recognition, the automatic recognition of unconstrained, natural handwriting today is still a difficult and scientific challenge. Automatic handwriting recognition performance profits only indirectly from technological advances such as increased computing power. The inherent variation of styles and the variability in a writer's behavior require (a) fundamental insight of the handwriting production process, (b) domain knowledge on the nature of script pattern geometry, and (c) powerful algorithms which display both noise tolerance and the ability to integrate multi-level information sources. The handwriting recognition research groups around the world are very small as compared to the effort spent in speech recognition. Still, the appeal of the idea that written words can be transformed into a neat machine-print font and can be handled by the computer is so strong, that university groups are trying to tackle this problem, again and again. Similarly, companies try to put forward pen-based computers, with limited or varying success. Why is it so difficult to translate the relatively simple idea of 'writing on a computer' into a reliable, easy and attractive system? It appears that the integration of pattern recognition modules into usable applications is far from trivial. The market failure of pen computing in the early nineties played an important role in motivating a reassessment of the pen-computing technology at a number of levels. In this paper, four different aspects of handwriting recognition and pen computing will be presented: Pen-computing hardware (section 2), Software and user interfaces (section 3), Handwriting: process and product (section 4), Recognition of on-line handwriting (section 5), followed by a concluding section.

[*]Paper, presented as keynote address at the IPA'97 , Thursday July 17th, Dublin, and appeared in IEE Electronics Communication Engineering Journal, 10(3), pp. 93-102, 1998.

2 Pen-computing hardware

The late sixties and early seventies witnessed the birth of a wide range of XY-position sensing devices. These transducers used either resistive, capacitive, electromagnetical, acoustical or pressure-sensitive technologies for the measurement of pen-tip position as a function of time. The technological developments allowed for an accurate planar position sensing such as was needed for graphical input in computer-aided design (CAD), especially in the automotive industry. Figure 1 gives a schematic description of the electromagnetic approach using tetherless pens.

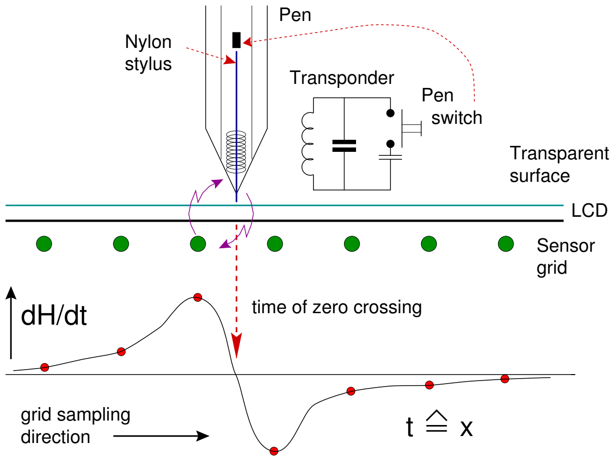

Figure 1: A schematic view of the electromagnetic transponder approach to pen-tip position sensing. A controller samples the field strength emitted by the resonating tuned circuit at each line of a relatively coarse grid. Low-pass filtering of the sensed signal strength followed by differentiation yields a good position estimate on the basis of the time of zero crossing. Other modern approaches are based on a pressure-sensitive writing surface.

Visionary ideas like Alan Kay's [7] Dynabook (1968) gave the impetus to a new hardware development: the integration of position-sensing technology with graphical display technology into a form of 'electronic paper' (EP). Early experiments involved standard CRT screens which were embedded in an office desk and equipped with some form of position sensing. In the middle of the eighties, the first real electronic paper prototypes appeared: The British National Physics Lab (NPL) produced a plasma display with integrated XY tablet, and IBM developed early prototypes of electronic paper. During the late eighties, the first integrated LCD/XY digitizers started to appear. These EP units were monochrome, and did not have back lighting (**1st generation EP**). Later in the early nineties, grey-level electronic paper devices started to appear. The presence of grey levels and of back lights (**2nd generation EP**) was a considerable improvement. However, the subsequent development of color EP was slow. In some cases the color LCD technology interfered with the accuracy of position sensing. But by 1995, there were several color LCD EP devices (**3rd generation EP**) available and in use in pen-based notebook computers. Note that the high graphical resolution and contrast of a ballpoint trace on plain white paper is unsurpassable with current electronic-paper devices. Today, we can make a distinction between three types of mobile pen computing platforms, from large to small:

1. pen-based notebook computers ('slates').
2. personal digital assistants (PDAs, 'handhelds'),
3. organizers ('palmtops'),

Since the early nineties it was realized that the presence of telecommunication functionality is an important aspect of hardware in small mobile information appliances. Without telecommunication, the added value of a mobile system is limited, reducing it to yet another and isolated information-processing environment for the user to handle. The integration of pen input and telecommunication was first demonstrated by the EO company with its Personal Communicator (1993). This device, which was very advanced for its time, contained all the basic ingredients which are considered necessary for mobile information processing and communication (fax, login and E-mail from wherever you are). Another very well-known example of a pen-based hand-held computer is the Apple Newton (1993). Although it was an innovative design with many new hardware solutions, its telecommunication functionality is limited. Later variants were aimed at solving this limitation, such as Motorola's Wireless 'Marco' Communicator (1995) which was based on Apple's Newton platform definition for PDAs. The last type of pen-based device to be mentioned is the small pen-based organizer. This type of device is gradually gaining wider acceptance, as witnessed by the popularity of the PalmPilot by 3COM.

From the hardware point of view, a keyboard is an expensive component of a mobile computer, and miniaturization quickly leads to sizes which are ergonomically unacceptable. The electronic-paper device, on the other hand, can be produced more efficiently in large quantities and allows for graphical input (including 'point and click') as well as handwriting recognition. However, until today, none of the above pen-based hardware designs proved to be attractive to a large audience, contrary to market expectations in the early nineties. The developments in the area of pen computing show that miniaturized mobile computer hardware containing a bag of functions, including pen input and an electronic paper display, does not automatically entail a useful, usable and attractive product.

3 Software and User Interfacing

The hardware history of pen computing is clearly characterized by a strong 'technology push'. But computer users will ask questions like: "What can I do with this device?"; "What is its added value with respect to the regular hand-held computer (mobile phone; pen and paper; fax)?"; "Do I have to learn a whole new computing environment?"; "Can I use my regular word processor?" In this section, we will see whether the developments in pen-based operating systems and applications are based on such essential user concerns.

The initial ideas on pen computers all revolved around the idea of the computer as a form of intelligent booklet, which was under the user's control through handwriting and pen gestures. During the eighties, the 'pen-and-paper' metaphor was explored by a number of research groups and companies. The idea of the computer as a direct replacement of paper was implemented by a number of companies in the form of an interface which looked like an active notepad with tabs, and was controlled by pen gestures. Today, the paper-mimicking approach has been abandoned to a large extent. There are several reasons for a growing disappointment with the paper and pencil metaphor. A pen computer is different from a piece of paper: Apart from the mere storage of information, the computer is a dynamic transformer of information, in many new ways which were not envisaged by the early visionaries. And, at times, the pen-computer-as-active-paper may act as a piece of rather uncooperative paper, for instance in the case of bad handwriting recognition results. Thus, there is a friction between the concept of paper, with its static properties, and the concept of a computer, as it has evolved in recent decades. In the case of paper, the writer is the only 'agent' who is fully in control of the graphical content. In a computer application, things are very much different. One thing which computers do, for instance, is cleaning up the human-generated information by representing it in regular grids of clearly rendered characters. Examples are the text editing and spreadsheet calculation software. The screen contents can be modified and reorganized very quickly, and the environment is totally different from the static environment of regular paper. Today, there exists a large population of computer users who are all very well accustomed to several paper-less forms of information processing. An example of a concept in the pen-and-paper metaphor is the notion that users should never have to **SAVE** a piece of information. This is based on the fact that on paper, objects are persistent once produced. The problem with this concept is that in current styles of computer use, the user will assume that there exists a clean version of a document on the hard disk, and a scratch version in working memory in which one may fiddle around until the document version is improved and worth to be saved. From this point of view, a rigorously pursued pen & paper metaphore means a step back.

Table 1: Currently available pen-computer applications

Application	*Description*	*Pen functions*
Form filling	office, marketing forms	tap, write, gesture
Phone operator functions	note taking, logging and forwarding	tap, write ink
Paint programs	art work	draw, write ink
Draw programs	technical drawing	draw, gesture, ink, HWR
CAD programs	technical drawing	draw, gesture, write
Note taking and editing	miscellaneous text entry	gesture, write ink, HWR
Insurance	reporting and decision support	gesture, write ink, HWR
Clip-on patient information systems	hospital, ambulance	tap/tick, write ink
Time-sheet data management	mobile workers	tap, gesture, write ink, HWR
Data collection and analysis tools	field geologic or biological data	tap, write ink
Building maintenance	e.g., roof, wiring inspection	tap, write ink
Architecture, home design	floor and yard plan diagramming	tap, draw
Fire-fighter operations	building maps and decision support	tap, write ink
Geographical information systems (GIS)	fast annotation on given GIS info	tap, write ink, HWR
Military field note taking	status reports, GIS annotation	write ink, tap
Routing and accounting	mobile work force	tap, write ink
Air traffic control	flight strip annotation	gesture, write ink
Pen-based web browsing	'using Netscape in the train'	tap

(HWR=handwriting recognition, tap=point and click with the pen)

The opposite approach to the pen & paper metaphore, however, is based on a generic window environment. It consists of replacing the computer mouse by a pen and adding some isolated handwriting recognition gadgets. This latter user-interface concept failed as well. A version of Microsoft Windows 3.1 was introduced around 1990 (PenWindows, later Windows for Pen Computing), dedicated to run on pen-based notebook computers, mainly. Both the handwriting recognition and user interfacing were suboptimal. As regards the user interface it was evident that the mouse could not be simply replaced by a pen. On the contrary, a total redesign of operating system and applications would be required. The disappointment with both approaches in Pen-User Interface (PUI) design has led many to believe that pen computing is inherently useless, which is actually an overreaction. Table 1 shows a list of application areas which are currently possible on pen-based computers. This table shows that, at least in specialized areas, there seems to be a use for the pen. However, in order to better understand the problems in pen-computer usability, it is useful at this point to take a look at some more fundamental issues.

The true bottleneck in human-computer interaction is not located in the computer-output to human-input channels. Provided that the information is presented in a structured way, the bandwidth of the channel from system to user can be rather high. However, as regards the human output to computer input, the bandwidth is very low. Speech has a reasonable bandwidth in symbolic terms: 100 words/min., while the average typing speed is 60 words/min. and handwriting is 20 words/min. (all rates depend on the language, in this case English). The spectral bandwidth of movements produced by the hand is about 10 Hz.

The miniaturization of mobile computers and their *keyboard* has a detrimental effect on typing speed. Also, *speech* cannot be used for all modes of computer interaction such as drawing and the editing of text. These observations imply that we need all the human-computer bandwidth we can get. Therefore, the *pen* cannot be dismissed carelessly, in spite of all problems encountered currently in the design and development of the pen-computing user interface. Other pointing devices, such as the track ball, joystick, and track point do not allow for accurate entry of alphanumerical or graphical symbols, or drawings. The mouse, contrary to the pen, is just a pointing instrument for selecting objects, and menu items. Muscles for coarse motor control are used in manipulating the mouse. Some continuous control like the dragging of screen objects can be performed with the mouse after some training but continuous control is not a strong point of the mouse. Also, drawing and sketching are very difficult. With a pen, however, the same actions can be performed as with a single-button mouse. The acuity of pen-tip positioning is very high because of the high number of degrees of freedom represented by the fingers, and the relatively large portion of the motor-control areas in the human brain which are dedicated to finger movement. There are additional functions which are typical for the pen: More elaborate *data entry* is possible, in the form of linguistic data (text) or graphical data (ink). Also, the user has a more direct control over objects on the screen: there is a higher degree of *Direct Manipulation*, similar to finger touch screens, but with a much higher spatial resolution in case of the pen. Table 2 presents a taxonomy of types of pen input which could be made useful in pen computing. A basic distinction in pen input is between (1) Textual Data Input, (2) Command Input, (3) Graphical Input. The topic of Signature Verification (4) has been covered recently in this journal[2]. For the purpose of the current paper, a number of aspects are relevant. First, it should be noted that many of the forms of input are very difficult to realize by either speech or a keyboard. Second, it is evident that handwriting recognition as such, is only a limited part of the required functionality in pen computing. In fact, it has become clear that 'electronic ink' alone as a data type has some very useful applications. Ink can be faxed easily, and the storage of notes together with time stamps and additional alphanumeric annotation can already be a large improvement over the use of paper for a given type of application. In case the end user of the pen-generated information is a human again, pattern recognition is often not necessary at all. The 'electronic ink' is a typical fundamental type of medium, differing from other basic media such as 'image', 'video' and 'sound'. An electronic-ink object has a number of typical properties. It refers to a pen-tip trajectory, possibly including pen angle and axial pen force, and it has a translation along the x and y axis, a scale, and a slant. Further rendering attributes are color, line thickness and brush. But the most important aspect of ink is that, like speech, it is a direct time function of human motor output. Ink can be simply displayed as an image, or played back in time. Unfortunately, 'electronic ink' is always forgotten in international data standards (e.g., ODA, ISO standard 8613:1989). A simple example of replaying recorded ink is provided by a tool that generates animated GIF images of a sample of handwriting[1].

As regards automatic recognition, it is a striking fact that the most natural form of handwriting input (Table 2: 1.1.1.1 - free text entry, unconstrained) cannot be handled well by current technology. This is both true for the pattern recognition functionality, as well as for the design of the user interface for free text entry. This problem is circumvented by using constraining form dialogs for the isolation of meaningful handwriting segments (Figure 2). Isolated characters (i.e., 'hand print', digits and block capitals) can be handled fairly well ($> 95\%$), isolated neatly written words can be recognized with a lower performance ($> 70\%$), and free text in mixed styles is still very difficult to handle.

Figure 2: Typical dialog box, prompting for isolated handwritten characters. Although the quality of elicited character shapes is improved, this mode of data entry may be slow and tedious, especially if the recognizer (still) does not classify the characters correctly. For each dialog box, advance knowledge on allowable input may be used by the handwriting recognizer to improve the classification accuracy.

[1]See: the free UNIPEN upTools3 package at http://hwr.nici.kun.nl/unipen/uptools3/

Table 2: A taxonomy of pen-based input

1 Textual Data Input

- **1.1** Conversion to ASCII
  - **1.1.1** Free Text Entry
    - **1.1.1.1** Fully unconstrained (size, orientation, styles) (e.g. PostIts)
    - **1.1.1.2** Lineated form, no prompting, free order of actions
    - **1.1.1.3** Prompted
      - **1.1.1.3.1** Acknowledge by "OK" dialog box
      - **1.1.1.3.2** Acknowledge by Time-out (e.g. 800 ms)
      - **1.1.1.3.3** Acknowledge by Gesture (see 2.3).
  - **1.1.2** Boxed Forms
  - **1.1.3** Virtual keyboard
- **1.2** Graphical text storage (plain handwritten ink)

2 Command Entry

- **2.1** Widget selection
- **2.2** Drag-and-drop operations
- **2.3** Pen gestures
  - **2.3.1** Position-independent gestures.
  - **2.3.2** Position-dependent context gestures.
- **2.4** Continuous control (e.g. sliders, ink thickness by pressure)

3 Graphical Pattern Input

- **3.1** Free-style drawings
- **3.2** Flow charts and schematics
- **3.3** Mathematical symbols
- **3.4** Music scores

4 Signature verification

Recognition technology also plays a role in a number of pen-input modes apart from numbers and text. For instance, research is being performed in the area of recognizing musical notation, and mathematical formulas [1]. Another research area concerns the beautification of flow charts and schematics [9]. A fundamental insight in the area of pen computing is the fact that the integration of a reasonable pattern recognition module (e.g., with a character recognition rate of 95%), into a usable application is very difficult. The main problem is that the recognition modules would have been designed differently, if there had been a clear focus of attention on the target application in the early development phase of the pattern recognizer. For example, who wants to write isolated characters, or isolated words? Users want to produce texts as fluently as possible. And when it comes to entering financial information, users demand 100% accuracy. The recognition of, e.g., natural writing behavior requires a different pattern recognition approach than the recognition of isolated words. Additionally, more often than not, pattern recognition problems can be solved by 'cheap tricks' in the user interface, like pop-up menus with recognition alternatives (Figure 3), on-screen virtual keyboards, and early recognition of characters to find a range of entries or records in an alphabetical database[3]. A virtual QWERTY keyboard can be presented on the LCD/digitizer and individual keys can be tapped with the pen. In fact reasonable speeds can be reached with the proper layout [14].

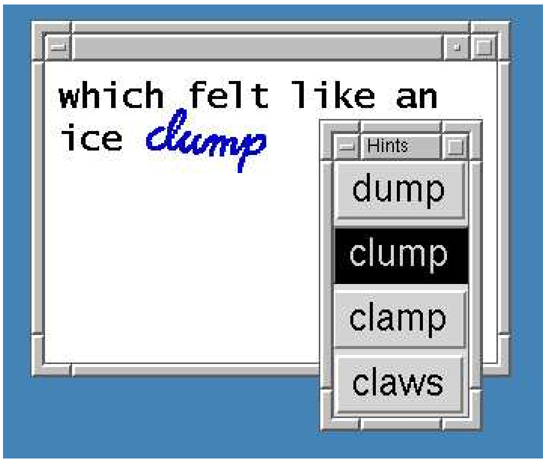

Figure 3: An auxiliary pop-up menu with recognized words as hints. Words are sorted from high likelihood (top) to low (bottom). The correct word (*clump*) happens to be in the second position and can be easily selected by the user to be entered into the working document.

Simplified alphabets, which make it easier for the recognizer are accepted by the end users, contrary to the predictions and expectations of many researchers. A new artificial style has been proposed by Goldberg, which has the advantage of being recognized with almost 100% accuracy [4]. The shape of these characters has nothing in common with the known alphabet (Figure 4). The symbols are entered in a small box on the screen, recognized by the machine and the machine font counterpart is put in the current text, after which the input symbol is erased. Several commercial implementations are now based on this idea (e.g. 'Graffiti'), mostly by providing for less extreme deviations from the regular alphabet, while maintaining the idea of a high separability of shapes for the classifier algorithm.

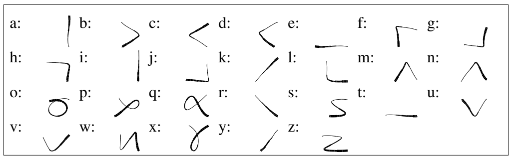

Figure 4: A simplified alphabet after Goldberg & Richardson [4], which makes things easier for the recognizer but necessitates human learning. Goldberg claimed it could be learned in 10 minutes. Indeed, in practice, a reasonable 20 min. are observed in motivated users.

New developments in the user interface

In the area of wearable computers, handwriting recognition modules can be re-used in finger gesture recognition. The user looks at a scene in the head-mounted display. Wearing a colored thimble, the finger movements and gestures can be processed and analyzed using similar algorithms as in on-line handwriting recognition.

A new and challenging area in which the NICI handwriting recognition group is currently active, is the annotation of image data by use of pen-based outlines. The new compression schemes MPEG-4 and MPEG-7 allow for an object-based description of images (as opposed to earlier rectangle-based schemes). This necessitates powerful tools for the creation of multimedial content. Objects have to be defined and annotated. The pen may prove to be a very useful tool in this area. Initial work in this direction already shows very promising results (Figure 5), integrating outline matching, image matching and semantic modeling within an information retrieval system.

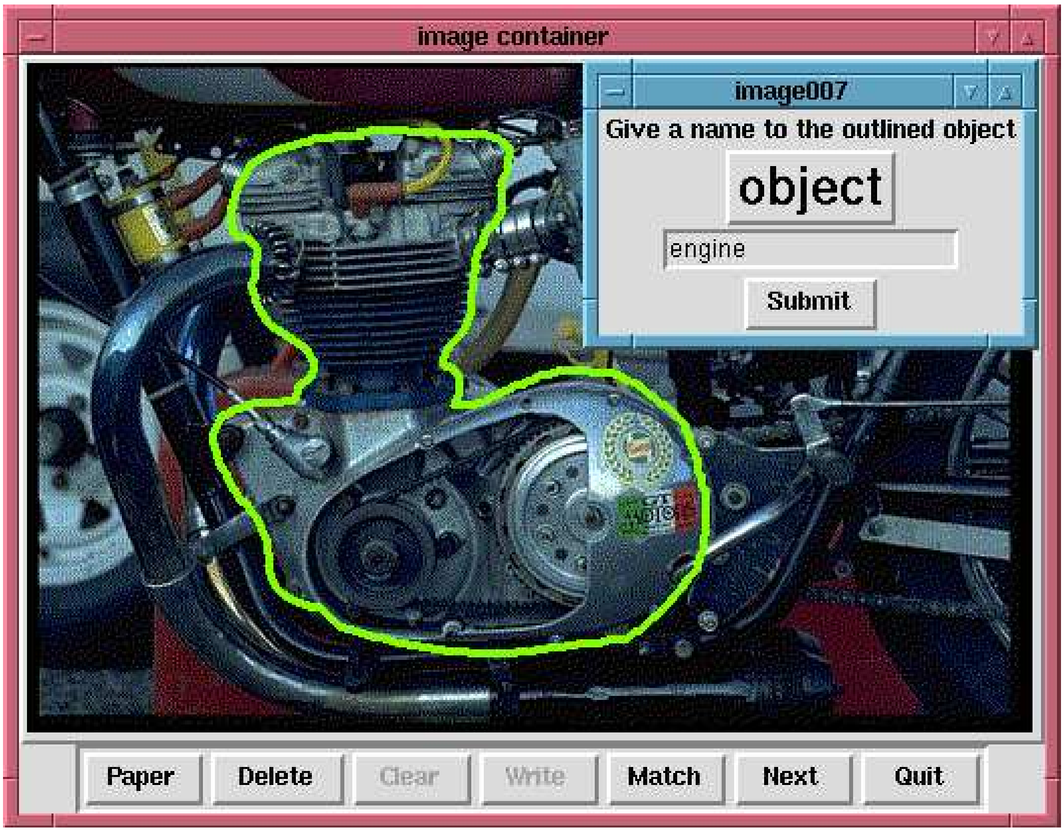

Figure 5: An example of Query By Image Content using the pen. In this example, an image sub-object has been outlined and annotated. Later queries can be based on either symbolic or pictorial matching or both. *Copyright 1997, NICI. Interface by J. Mackowiak, matching algorithm L. Schomaker*

Finally, it should be noted that a number of activities in the real world are still done with pen and paper, and are a potential candidate for pen-computing applications. The storyboards in movie, multimedia, and computer game production are still mostly produced with pen and paper. Even in the area of user-interface design itself, the pen is often a preferred tool.

4 Handwriting process and product

Handwriting and *Drawing* are two different means of human information storage and communication, produced by the same single two-dimensional output system: a pointed writing implement, usually driven by the hand and arm, which leaves a visible trace on a flat surface. *Handwriting* conveys symbolical data, whereas *Drawing* conveys pictorial data. There exists a third data type, *Pen Gestures*, consisting of unique symbols, as used, e.g., traditionally by book editors and in pen computers. A gesture is a non-alphanumeric symbol which, when produced, requires a given function to be executed.

Contrary to speech, handwriting is not an innate neural function, and must be trained over several years. During the training process, handwriting evolves from a slow feedback process involving active attention and eye-hand coordination to a fast automatic and ballistic process. The atomic movement unit in handwriting is a *stroke*, which is a movement trajectory bounded by two points of high curvature and a corresponding dip in the tangential movement velocity (Figure 6).

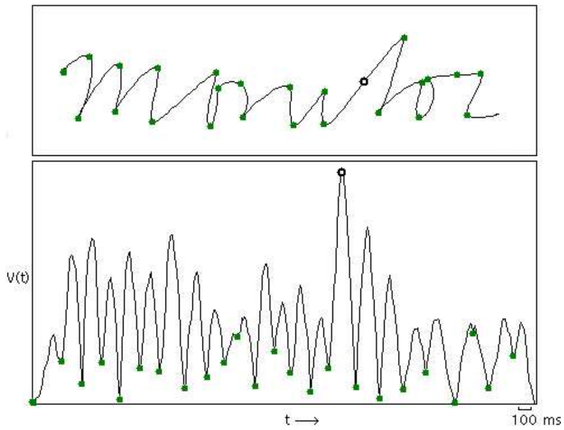

Figure 6: A cursive-written word *(monitor)* and the time function of the corresponding pen-tip velocity. Note that points of high curvature are characterized by a dip in the velocity. A trajectory between two velocity minima is called a 'stroke'

The typical modal stroke duration is of the order of 100 ms, and varies much less for increased movement amplitudes than one would expect: For a large range of amplitudes, writers exert an increased force in order to maintain a preferred rhythm of movement (10 strokes/s, 5 Hz). Once fired cortically, at a high level in the brain, such a stroke cannot be corrected by visual feedback, hence it is considered to be a ballistic phenomenon. The handwriting process evolves in a continuous process of concurrent advance planning and real-time execution, the planning process being 2–3 characters in advance of the execution process. Writing errors convey the fact that during the writing, several processes take place at the same time, including phonemic-to-graphemic conversion (spelling), and graphemic-to-allographic conversion, i.e., the choice of letter shapes (say, "font choice"). Given the complexity of the handwriting task, one may understand that it is a rather sensitive process, vulnerable to external disturbances. These influences are either of a pharmacological (alcohol, drugs), or of a cognitive nature (noise, distractions, movement, talking of other people in the direct environment). Handwriting requires a higher degree of selective attention than speech. But the fact is that many humans have enjoyed ample training experience in handwriting and drawing, and are thus able to produce accurate and small movements, if needed. Table 3 gives an overview of parameters that may be controlled with a pen.

Table 3: Parameters controlled by a pen

Parameter	*Description*
x, y	position (velocity, acceleration...)
p	pen force ("pressure"):
	- through a binary pen-up/pen-down switch
	- through an analog axial-force transducer
z	height of the pen tip w.r.t. the table plane
ϕ_x, ϕ_y	angles of the pen w.r.t. the table plane
Switching	i.e., by thresholding of pen force p
	or based on additional button(s) on the pen

5 Recognition of on-line handwriting

Several authors have already produced excellent overview papers on handwriting recognition. In the area of off-line recognition (i.e., pixel-based spatial input representations), there is a paper by Suen et al.[12]. The input signal is usually described by a grey-scale function $I(x, y)$. In the area of on-line recognition (i.e., vector-based spatiotemporal input representations) there is an extensive overview by Tappert et al. [13]. Here, the input signal consists of a sequence of vectors (x_k, y_k). Although there have been many new developments since, these papers offer a good introduction. The advantage of the pen-computer platform is that both representations $I(x, y)$ and (x_k, y_k) can be used, whereas handwriting recognition based on optical scanning is largely based on $I(x, y)$. In this paper, the focus will be directed at on-line recorded handwriting.

The main problem of automatic handwriting recognition is the search for invariance, i.e., for methods which reduce the variation and variability in the input. Figure 7 is a schematic description of the four basic types of variation and variability which have to be solved.

The first category of variation in the handwriting input signal concerns the **Affine transforms** that the writer imposes on the handwriting (Figure 7, A). Translation, scale, shear and rotation can be varied by the writer at will, within limits. For the removal of affine geometrical variations, a number of methods can be designed, usually based on linear algebra. Sometimes, the affine normalization is applied locally, in a sliding window, and is then called a local-affine transform. As an example, shear can be normalized by estimating handwriting slant and normalizing to a given standard slant. Alternatively, a linear system estimation can be performed to minimize the distance between a character template and the current input window. The resulting minimum distance is then used in the character classification.

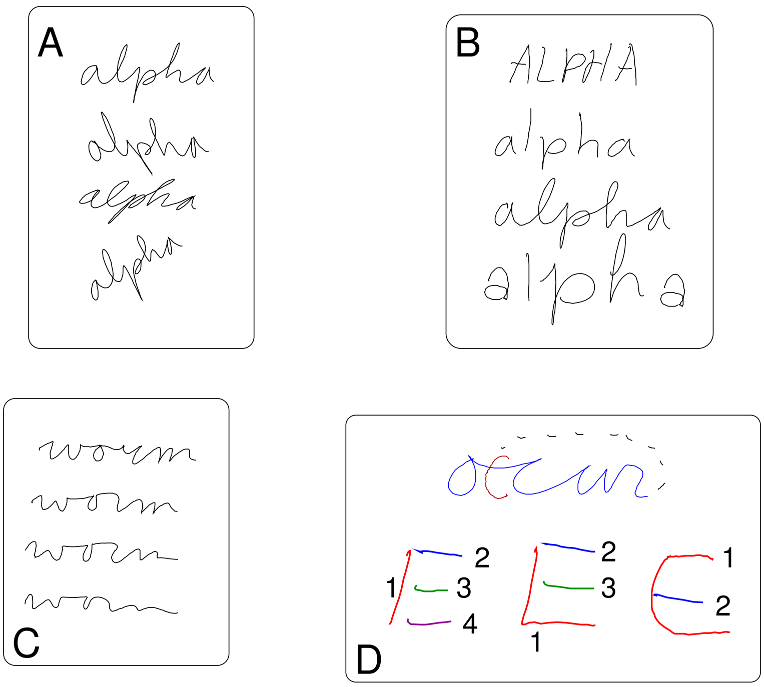

Figure 7: Four basic sources of variation and variability in handwriting. (A) Affine transforms, (B) Allographic variation, (C) Neuro-biomechanical variability, and (D) Sequencing variability. Handwritten examples are given. A robust recognition algorithm needs too solve the problems in all four areas.

The second source of variation concerns **Allographic variation**, or the amount of character shapes used within the writer population for a given letter in the alphabet (Figure 7 B). There are large shape differences between characters produced by different writers, especially when they are of different nationality, of different generations, or if they were taught different writing style methods. This variation is the toughest problem in handwriting, and the main reason for the initiation of the UNIPEN[5] project in which a database on Western handwriting styles has been collected by over 40 companies and institutions. Many pattern recognition techniques try to solve this problem by blind massive training, as in the case of multi-layer perceptrons or hidden-Markov recognizers. Such classifiers run the risk of generating average, but incomplete representations of the total ensemble of allographic variants. Figure 8 shows hierarchical clustering results for a set of 1800 randomly drawn characters, 600 *g*, 600 *f* and 600 *k*. Without manual intervention, the system [16] has detected a family tree of letter shapes (allographs). The ultimate goal is to provide for a more systematic naming of shapes. After all, different machine-print font names are used for shapes differing in only a few pixels per character, whereas in handwriting recognition there is no agreed naming scheme for the diverging character shapes.

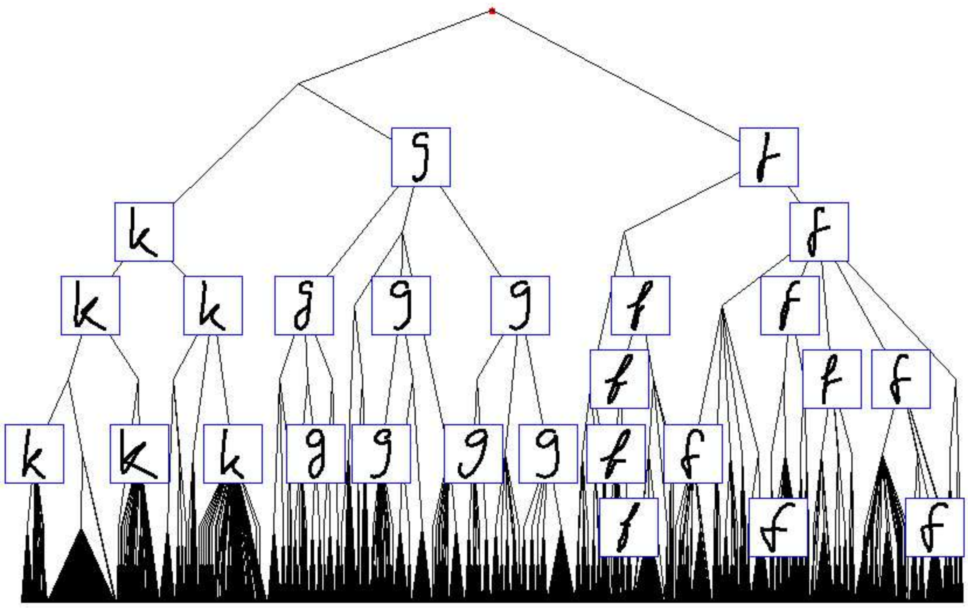

Figure 8: A 'family tree' of character shapes for *k, g* and *f*, as obtained by a new variant of hierarchical clustering developed at NICI [16]. The top node represents all 1800 input characters, whereas the nodes on the bottom line are the individual character samples (members). An allograph within a rectangle is the simple average $(\hat{\vec{x}}, \hat{\vec{y}})$ of all members.

In general, every new writer will produce some variation, ligature or curl in the allographs which is not yet present in the training set: Although the UNIPEN data set is now five million characters, this is apparently not enough. In the seventies and early eighties, it was hoped that the problem could be solved by identifying universal structural features (e.g. *rule #654: "all t's are crossed"*), but this did not work due to the huge style variation. Not all t's are crossed in real life.

The third source of problems in automatic recognition of handwriting, **Neuro-biomechanical variability**, comes from the neurophysiological and biomechanical limitations of the human writing apparatus, such as bandwidth and processing speed and quality (Figure 7 C). For instance, speeding up writing means that more force should be produced for maintaining the target curvature values, in less time. If this fails, handwriting becomes sloppier in a characteristic manner. There have been some attempts to solve this problem by local deconvolution of the signal. Other researchers have tried to explore the 'sloppiness space', using autoregressive models [8] or Fourier descriptors [6]. Since some forms of sloppiness occur frequently and may be the same for several writers, big training sets will have a beneficial effect here, too. If explicit allograph templates are known, the standard deviations of features in a feature vector are useful as an estimate of the 'sloppiness space'. In human motor control, a distinction can be made between articulatory processes ('*shaping*'), and concatenation processes ('*chaining*') [10]. The variability problems which are meant here, fall in the *shaping* category, while problems in *chaining* are categorized in the next paragraph.

The fourth source of variation, **Sequence variability**, refers to the variable order in which handwriting may be produced (Figure 7 D). Post-hoc editing, crossing of t-bars, dotting of the i's and j's are typical. Also, spelling errors cause a problem, which are due to limitations in the writer's linguistic knowledge. Then there are the slips of the pen: Letter omissions and insertions which have their origin in motor-control processes. On-line handwriting recognition may suffer more from this type of variability than off-line image-based recognition, although many forms of retracing and the resulting overlap of shapes will decrease the off-line classifier recognition rates as well. Current technology does not deal well with this problem. Even if there are severe spelling errors or letter omissions, the human writers and readers often identify the nearest lexical match where the algorithm produces a *reject* response at best. Another paradigmatic example is the block print capital E. If the E is produced in four strokes, each of which can be started at two ends, this results in $2^4 * 4! = 384$ sequence variants. Luckily, human writers do not actually perform all of these permutations, and restrain themselves to a limited subset. To some extent, the problems in this category can be solved by stroke reordering, as is usually done in recognizers of Chinese (Hanzi), Japanese (Kanji, Katakana, Hiragana) and Korean (Hangul) scripts.

Input categories using handwriting

Hierarchically, handwriting consists of the following components, from small to large: *stroke, character, word, sentence, paragraph, and page*. In practice, only the levels from stroke up to word are well known. Only a limited number of researchers have explored the sentence and paragraph levels. Like in any other pattern recognition system, the object definition precedes and determines the feature vector definition. Table 4 shows the basic input object definitions in the case of handwriting recognition:

Table 4: Categories of handwriting shapes

Style	*Description*
I	digits
II	block capitals
III	isolated handprint
IV	run-on handprint/mixed-cursive words
V	fully connected cursive words
VI	punctuations
VII	gestures, markup symbols
VIII	free text, combinations of I-VII.

Typically, these different input categories are tackled by different types of algorithms. The performance of dedicated digit recognizers is likely to be much higher than the performance of a word-oriented recognizer which includes digits as a possible input class. It should be noted that users do not make such a distinction as given in Table 4 and will not understand if they are not allowed to write digits into a dialog box which is determined for the recognition of isolated words. Similarly, the mixed use of capitals and connected-cursive script is difficult to handle, and must be realized by integrating the results of multiple classifiers which look at the same input pattern. Apart from this given coarse distinction in input categories many more subtle style variations exist, as was already shown in figure 8. The design of algorithms for combining information from several classifiers plays a central role in current research. The following list summarizes current basic pattern recognition methods:

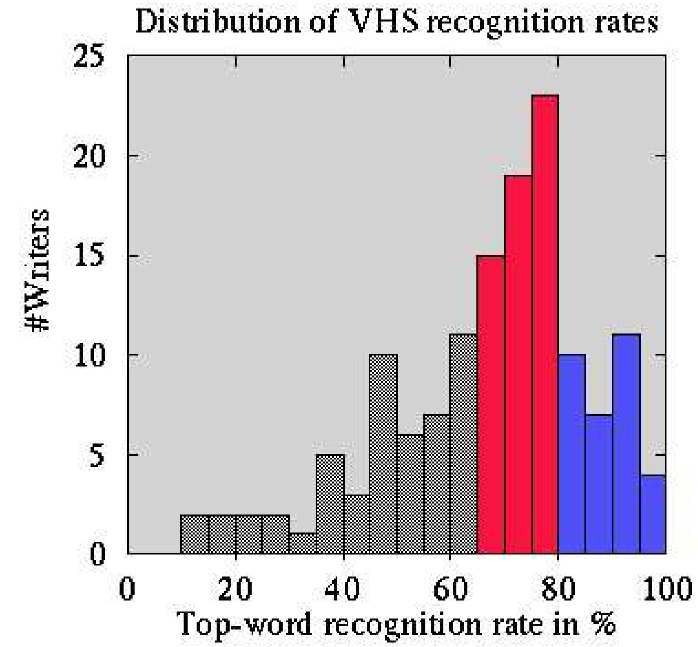

Figure 9: A distribution of test set recognition rates of one, older version of the VHS handwriting recognizer. Note, that for some writers, the performance is quite acceptable, whereas the character shapes for others (left tail of the distribution) are insufficiently represented within the system.

- rule-based methods using structural character features and decision trees,
- artificial neural networks (ANN) such as the multi-layer perceptron and Kohonen self-organizing feature maps,
- traditional statistical pattern recognition methods such as discriminant analysis,
- hidden Markov models.

In a multiple-classifier scheme, several of these techniques may be used in parallel, each of them addressing different major categories of ink input. The combination methods which are involved are in the same ballpark as those used in multisensor data fusion[15].

A stroke-based recognizer of on-line handwriting

The NICI stroke-based recognizer of on-line handwriting [11] was developed on the basis of knowledge on the handwriting production process. Because it acts as an inverse transform of the human motor product, yielding the intended word identity, it was dubbed Virtual Handwriting System (VHS)[2]. Assuming equidistant sampling in time, the basic component of the handwriting signal is in our view the velocity-based stroke (VBS). This approach is attractive because of the strong coupling between the curvature of the trace and the tangential velocity. The majority of the writers produce ballistic movements without too many hesitations or other accidents, especially in connected-cursive script. The approach is not suited for children's handwriting or handwriting with tremor.

Recognition performance measures should be interpreted with extreme caution. The rates are seldomly underestimated in literature. Figure 9 gives a distribution of recognition rates in an *unseen group of writers* for the stroke-based recognizer ('95 version). It is the top-word recognition rate of the word classifier: "How often was the system's best guess indeed correct". No word-shape information or

[2]A Web page is present at http://hwr.nici.kun.nl/unipen/nici-stroke-based-recognizer.html

linguistic statistics were used. The recognizer just performs a strict search for individual letters, and all letters must be found. This means that all fused letters and spelling errors will lead to a missed word. Lexicon size was 250 words, each writer wrote 45 words. The average processing time on a HP-UX 9000/735 workstation is 215 ms per word. Note that when this system meets unseen writers, a substantial part of them will have low recognition rates. For example, some of the writers will write small 'all-caps' letters, claiming that such is their lower case handwriting. The VHS recognizer is only one of several methods we have tried over the last few years. Initially started as a pure connected-cursive recognizer, the approach gradually allowed for incorporating mixed handwriting and isolated handprint, as well.

6 Conclusion

In this paper, problems of handwriting recognition and the development of applications for pen computers have been addressed. Despite all problems, the use of the pen as an input device has survived, at least in small niches of the world of computer applications. Miniaturization of computers necessitates a rethinking of human-computer interaction, in which the pen may still cover a substantial portion of the human-computer channel bandwidth. Very often, a pen-based application will not require pattern recognition at all, relying instead on recording and rendering notes in electronic ink format. New applications are emerging in multimedia, such as image-based database queries, in which the pen may play a new and useful role. As regards the necessary improvement in handwriting recognition, current research topics are multiple-classifier integration, huge training sets (UNIPEN), and hidden-Markov modeling. Within our own research group, the current focus of attention is the development of allograph taxonomies using hierarchical clustering methods.

References

[1] A. BELAID and J. HATON. A syntactic approach for handwritten mathematical formula recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol PAMI-6 No 1:pp 105–111, January 1984.

[2] M.C. FAIRHURST. Signature verification revisited: promoting practical exploitation of biometric technology. *Electronics & Communication Engineering Journal*, 9(6):245–253, 1997.

[3] C. FRANKISH, R. HULL, and P. MORGAN. Recognition accuracy and user acceptance of pen interfaces. In *Proceedings of the ACM CHI'95*, pages 503–510. Addison-Wesley, 1995.

[4] D. GOLDBERG and C. RICHARDSON. Touch-Typing with a Stylus. In *InterCHI '93 Conference Proceedings*, pages 80–87, Amsterdam, 1993.

[5] I. GUYON, L. SCHOMAKER, R. PLAMONDON, R. LIBERMAN, and S. JANET. Unipen project of on-line data exchange and recognizer benchmarks. In *Proceedings of the 12th International Conference on Pattern Recognition, ICPR'94*, pages 29–33, Jerusalem, Israel, October 1994. IAPR-IEEE.

[6] S. IMPEDOVO, B. MARANGELLI, and A.M. FANELLI. A fourier descriptor set for recognizing nonstylized numerals. *IEEE Transactions on Systems, Man and Cybernetics*, SMC 8(8):640–645, August 1978.

[7] A. KAY. *User Interface: A Personal View. The art of human-computer interface design.* Addison Wesley, Reading, Massachusetts, 1990.

[8] S. KONDO. A model of the handwriting process and stroke-structure of character-figures. In *Computer recognition and human production of handwriting*, pages 103–118. World Scientific, 1989.

[9] J. PAVLIDIS and C.J. VAN WYK. An automatic beautifier for drawings and illustrations. *A.C.M. Computer Graphics*, Vol 19 No 3:225–234, July 1985.

[10] L.R.B. SCHOMAKER. A neural-oscillator model of temporal pattern generation. *Human Movement Science*, 11:181–192, 1992.

[11] L.R.B. SCHOMAKER. Using Stroke- or Character-based Self-organizing Maps in the Recognition of On-line, Connected Cursive Script. *Pattern Recognition*, 26(3):443–450, 1993.

[12] C.Y. SUEN, M. BERTHOD, and S. MORI. Automatic recognition of handprinted characters - the state of the art. *Proceedings of the IEEE*, 68(4):469–484, April 1980.

[13] C. C. TAPPERT, C. Y. SUEN, and T. WAKAHARA. The State of the Art in On-line Handwriting Recognition. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 12:787–808, 1990.

[14] T. VAN GELDEREN, A. JAMESON, and A. L. DUWAER. Text recognition in pen-based computers: An empirical comparison of methods. In *InterCHI '93 Conference Proceedings*, pages 87–88, Amsterdam, 1993.

[15] P.K. VARSHNEY. Multisensor data fusion. *Electronics & Communication Engineering Journal*, 9(6):245–253, 1997.

[16] L. VUURPIJL and L. SCHOMAKER. Finding structure in diversity: A hierarchical clustering method for the categorization of allographs in handwriting. In *Proceedings of the 1997 Fourth International Conference on Document Analysis and Recognition (ICDAR'97)*, pages 387–393. ICDAR, August 1997.

Figure 1.
A schematic view of the electromagnetic transponder approach to pen-tip position sensing. A controller samples the field strength emitted by the resonating tuned circuit at each line of a relatively coarse grid. Low-pass filtering of the sensed signal strength followed by differentiation yields a good position estimate on the basis of the time of zero crossing. Other modern approaches are based on a pressure-sensitive writing surface.

Figure 2.
Typical dialog box, prompting for isolated handwritten characters. Although the quality of elicited character shapes is improved, this mode of data entry may be slow and tedious, especially if the recognizer (still) does not classify the characters correctly. For each dialog box, advance knowledge on allowable input may be used by the handwriting recognizer to improve the classification accuracy.

Figure 3.
An auxiliary pop-up menu with recognized words as hints. Words are sorted from high likelihood (top) to low (bottom). The correct word (*clump*) happens to be in the second position and can be easily selected by the user to be entered into the working document.

Figure 4. A simplified alphabet after Goldberg & Richardson [4], which makes things easier for the recognizer but necessitates human learning. Goldberg claimed it could be learned in 10 minutes. Indeed, in practice, a reasonable 20 min. are observed in motivated users.

Figure 5. An example of Query By Image Content using the pen. In this example, an image sub-object has been outlined and annotated. Later queries can be based on either symbolic or pictorial matching or both. *Copyright 1997, NICI. Interface by J. Mackowiak, matching algorithm L. Schomaker*

Figure 6. A cursive-written word *(monitor)* and the time function of the corresponding pen-tip velocity. Note that points of high curvature are characterized by a dip in the velocity. A trajectory between two velocity minima is called a 'stroke'

Figure 7. Four basic sources of variation and variability in handwriting. (A) Affine transforms, (B) Allographic variation, (C) Neuro-biomechanical variability, and (D) Sequencing variability. Handwritten examples are given. A robust recognition algorithm needs too solve the problems in all four areas.

Figure 8. A 'family tree' of character shapes for *k, g* and *f*, as obtained by a new variant of hierarchical clustering developed at NICI [16]. The top node represents all 1800 input characters, whereas the nodes on the bottom line are the individual character samples (members). An allograph within a rectangle is the simple average $(\hat{\vec{x}}, \hat{\vec{y}})$ of all members.

Figure 9. A distribution of test set recognition rates of one, older version of the VHS handwriting recognizer. Note, that for some writers, the performance is quite acceptable, whereas the character shapes for others (left tail of the distribution) are insufficiently represented within the system.