From handwriting analysis to pen-computer applications

by L. Schomaker

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 viewpoints. Firstly a brief overview of the hardware developments in pen systems is given, leading to the conclusion that the technological developments in this area have not led to the expected user acceptance of pen computing. The reasons underlying this market failure are explored. Problems of pen-user interface design are then described and existing and new applications are summarised. The handwriting process and product are discussed and, finally, automatic recognition methodologies are considered. Four basic factors determining handwriting variation and variability are identified. A handwriting recognition approach using segmentation into velocity-based strokes is considered in somewhat more detail.

1 Introduction

Handwriting recognition and pen computing are characterised by an arduous evolution history. Originally identified 30 years ago as a first step towards the more difficult problem of speech recognition, the automatic recognition of unconstrained, natural handwriting is today 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 behaviour 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 multilevel 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 handled by a 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 pen-computing technology at a number of levels. In this paper, four different aspects of handwriting recognition and pen computing are presented: pen-computing hardware, software and user interfaces, the handwriting process and product and recognition of on-line handwriting.

2 Pen-computing hardware

The late sixties and early seventies witnessed the birth of a wide rang of *XY*-position-sensing devices. These transducers used either resistive, capacitive, electro-magnetic, acoustic or pressure-sensitive technologies for the measurement of pen-tip position as a function of time. The technological developments enabled accurate planar position-sensing, such as was needed for graphical input in computer-aided design (CAD), especially in the automotive industry. Fig. 1 is a schematic diagram of the electromagnetic approach using tetherless pens.

Visionary ideas, such as Alan Kay's' Dynabook (1968), gave the impetus to a new hardware development: the integration of position-sensing technology with graphical display technology to provide a form of 'electronic paper' (EP). Early experiments involved standard cathode ray tube (CRT) screens, which were embedded in an office desk and equipped with some form of position sensing. In the mid eighties, the first real electronic-paper prototypes appeared: the British National Physical Laboratory (NPL) produced a plasma display with an integrated *XY* tablet, and IBM developed early prototypes of electronic paper.

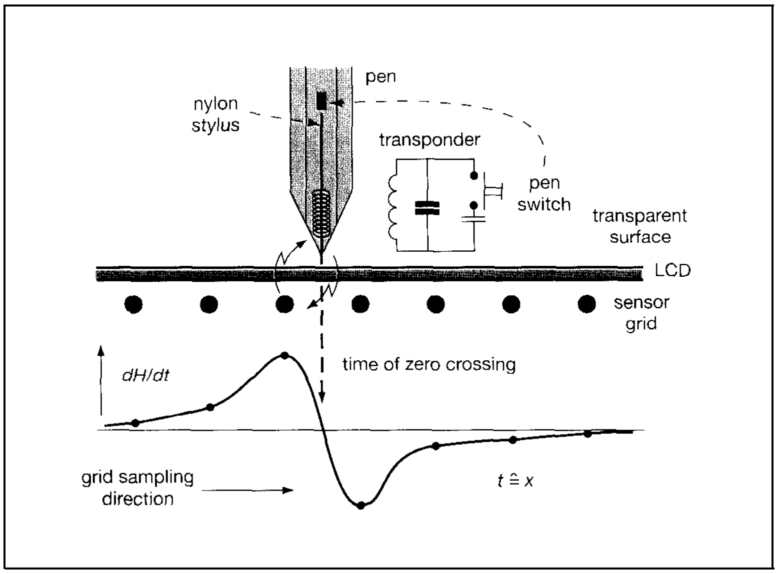


Fig. 1 **Schematic diagram of the electromagnetic transponder approach to pen-tip position sensing. A controller samples the field strength emitted by the resonating tuned circuit at each line of a relatively coarse grid. Low-pass filtering of the sensed signal strength followed by differentiaton yields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

During the late eighties, the first integrated LCD/*XY* digitisers started to appear. These EP units were monochrome and did not have back lighting (first generation EP). Later, in the early nineties, grey-level electronic-paper devices were realised. The presence of grey levels and of back lights (second generation EP) was a considerable improvement. However, the subsequent development of colour EP was slow. In some cases the colour LCD (liquid crystal display) technology interfered with the accuracy of position sensing. Rut by 1995, there were several colour LCD EP devices (third 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:

(a) pen-based notebook computers ('slates')
(b) personal digital assistants (PDAs, 'handhelds')
(c) organisers ('palmtops').

Since the early nineties it has been realised that the presence of telecommunication functionality is an important aspect of hardware in small mobile information appliances. Without telecommunications, 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 telecommunications 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 handheld computer is the Apple Newton (1993). Although it was an innovative design with many new hardware solutions, its telecommunication functionality is limited. Later variants, such as Motorola's Wireless 'Marco' Communicator (1995), which was based on Apple's Newton platform definition for PDAs, were aimed at solving this limitation. The last type of pen-based device mentioned above is the small pen-based organiser. 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 miniaturisation quickly leads to sizes which are ergonomically unacceptable. An 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, to date, none of the above pen-based hardware designs has 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 miniaturised mobile computer hardware containing a bag of functions, including pen input and an electronic-paper display, does not automatically result in a useful, usable and attractive product.

3 Software and user interfacing

The hardware history of pen computing is clearly characterised by a strong 'technology push'. But computer users ask questions such as: 'What can I do with this device?'; 'What is its added value with respect to the regular handheld computer (mobile phone, pen and paper, fax)?'; 'Do I have to learn a whole new computing environment?' 'Can I use my regular word processor?' This section examines the question of 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 to 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 pen-and-paper metaphor.

A pen computer is different from a piece of paper. Apart from merely storing information, a computer transforms

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 desision 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 geological or biological data	tap, write ink
Building maintenance	e.g. roof, wiring inspection	tap, write ink
Architecture, home design	floor and yard plans	tap, draw
Fire-fighting 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

*Tap = point and click with the pen
†HWR = handwriting recognition

information dynamically in many new ways which were not envisaged by the early visionaries. Also, at times, a pen computer may act as a piece of rather uncooperative paper, for instance in the case of bad handwriting-recognition results. Thus, there is a divergence between the concept of paper, with its static properties, and the concept of the 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 different. For example, computers can clean up human-generated information by representing it in regular grids of clearly rendered characters, and text-editing and spreadsheet calculation software enable the screen contents to be modified and reorganised very quickly. The computer environment is thus totally different from the static environment of regular paper. Today, there exists a large population of computer users who are all very familiar with several paperless 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 objects on paper are persistent once produced. The problem with this concept is that in current styles of computer use, the user assumes that there exists a clean version of the document on the hard disk and a scratch version in working memory which may be repeatedly modified until a version worth saving is produced. From this point of view, a rigorously pursued pen-and-paper metaphor means a step back.

The opposite approach to the pen-and-paper metaphor 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 type of user-interface concept failed as well. Aversion of Microsoft Windows 3.1 was introduced around 1990 (Pen Windows, later Windows for Pen Computing), mainly intended to run on pen-based notebook computers. Both the handwriting recognition and user interfacing were suboptimal. As regards the user interface it was evident that the mouse could not simply be replaced by a pen; a total redesign of operating system and applications would be required.

The disappointment with both approaches to pen-user interface (PUI) design has led many to believe that pen computing is inherently useless, a view which is actually an overreaction. Table 1 shows that, at least in specialised areas, there seems to be a use for the pen. However, in order better to 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; the average typing speed is 60 words/min and text can be handwritten at 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 miniaturisation of mobile computers and their keyboard has a detrimental effect on typing speed. Also, speech cannot be used for some modes of computer interaction such as drawing and the editing of text. These observations imply that we need all the human-to-computer bandwidth we can get. Therefore, the pen cannot be dismissed out of hand, in spite of all the problems that have been encountered 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 permit accurate entry of alphanumerical characters,

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' dialogue 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

graphical symbols, or drawings. The mouse, in contrast to the pen, is just a pointing instrument for selecting objects and menu items. Muscles for coarse motor control are used in rnanipulating the mouse. Some continuous control, such as dragging 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 mouse. The same actions can be performed with a pen as with a single-button mouse, however the acuity of pen-tip positioning is very high because of the high number of degrees of freedom provided by the fingers and the relatively large portion of the motor-control areas in the human brain which are dedicated to finger movement. A pen typically provides additional functions. For example, 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 the case of the pen.

Table 2 presents a taxonomy of types of pen input. A basic distinction in pen input is between textual data input, command input, graphical input and signature input. Signature verification was the subject of a recent paper" in this journal. For the purpose of the current paper, a number of aspects are relevant. Firstly, it should be noted that many of the forms of input are very difficult to realise by either speech or a keyboard. Secondly, it is evident that handwriting recognition is only a limited part of the required functionality in pen computing.

In fact, it has become clear that 'electronic ink alone is a data type with very useful applications. Ink can be faxed easily, and the storage of notes together with time stamps and additional alphanumeric annotation can be a large improvement over the use of paper for certain applications. If the end user of the pen-generated information is also a human, pattern recognition is often not necessary. 'Electronic ink is a fundamental type of medium, differing from other basic media such as images, 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 axes, a scale, and a slant. Further rendering attributes are colour, 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 displayed simply as an image, or played back in time. Unfortunately, 'electronic ink' is always forgotten in international data standards (e.g. ODA, IS0 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*.

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 true both 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 dialogues for the isolation of meaningful handwriting segments (Fig. 2). Isolated characters (i.e. 'hand printed' characters, digits and block capitals) can be handled fairly well (>95% recognition accuracy); isolated, neatly written words can be recognised with a lower performance (>70%); but free text in mixed styles is still very difficult to handle.

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 recognising musical notation, and mathematical formulas[3]. Another research area concerns the beautification of flow charts and schematics[4]. A fundamental fact in the area of pen computing is 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. Who, for example, 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, for example, natural writing behaviour requires a different approach to pattern recognition than the recognition of isolated words.

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

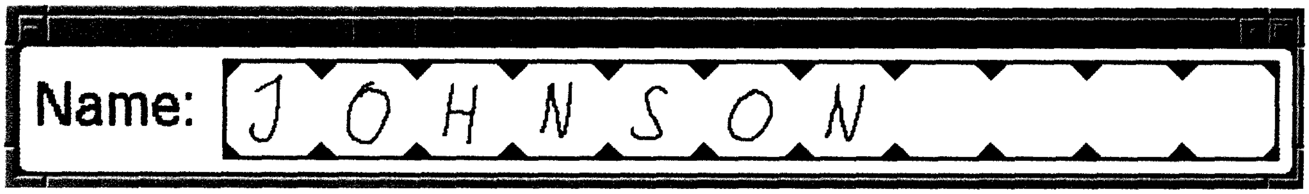


Fig. 2 **Typical dialogue box, prompting for isolated handwritten characters. Although better quality character shapes are elicited, this mode of data entry may be slow and tedious, especially if the recogniser (still) does not classify the characters correctly. For each dialogue box, advance knowledge of allowable inputs may be used by the handwriting recogniser to improve the classification accuracy**

Additionally, more often than not, pattern recognition problems can be solved by ‘cheap tricks’ in the user interface, such as pop-up menus with recognition alternatives (Fig. 3), on-screen virtual keyboards, and early recognition of characters to find a range of entries or records in an alphabetical database”. A virtual QWERTY keyboard can be presented on an LDC/digitiser and individual keys can be tapped with the pen. In fact, reasonable speeds can be reached with the proper layout”.

Simplified alphabets, which make it easier for the recogniser, are accepted by 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 recognised with almost 100% accuracy[7]. The shape of these characters has nothing in common with the known alphabet (Fig. 4). The symbols are entered in a small box on the screen, recognised by the machine and the machine font counterpart 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 less extreme deviations from the regular alphabet while maintaining the idea of a high separability of shapes for the classifier algorithm.

New developments in the user interface

In the area of wearable computers, handwriting-recognition modules can be reused in finger gesture recognition. The user looks at a scene in a head-mounted display and wears a coloured thimble. Finger movements and gestures can be processed and analysed using algorithms similar to those used for on-line handwriting recognition.

A new and challenging area, in which the Nijmegen Institute for Cognition and Information (NICI) handwriting recognition group is currently active, is the annotation of image data by using 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 multimedia 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 (Fig. 5), integrating outline matching, image matching and semantic modelling within an information retrieval system.

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, for example, 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 learned over several years. During the learning 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

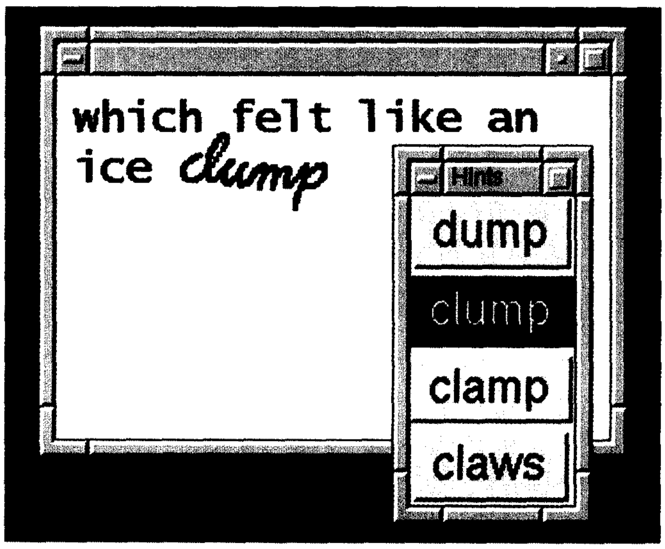
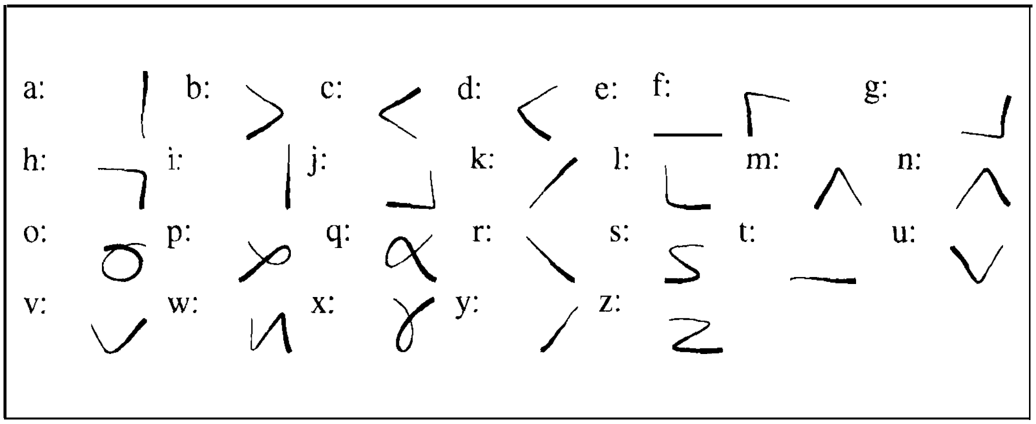


Fig. 3 **An auxiliary pop-up menu with recognised words as hints. Words are sorted from high likelihood (top) to low (bottom). The correct word (clump) happens to be in the second position and can be easily selected by the user to be entered into the working document**

Fig. 4 A simplified alphabet after Goldberg and Richardson’, which makes things easier for the recogniser but necessitates human learning. Goldberg claimed it could be learned in 10 minutes. In practice, about 20 minutes is required by motivated users



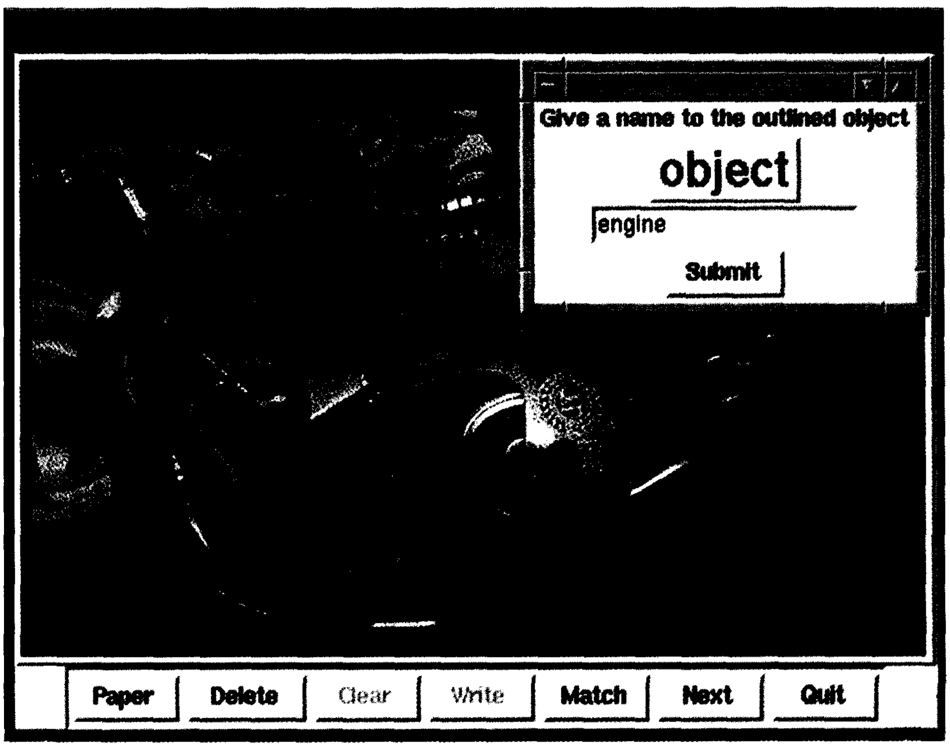
curvature and a corresponding dip in the tangential movement velocity (Fig. 6).

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, other people talking 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.

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.*[8] 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 Tapper-t *et al.*[9] 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

Fig. 5 An example of ‘query by image content’ using the pen. In this example, an image subobject has been outlined and annotated. Later queries can be based on either symbolic or pictorial matching or both (© NICI, 1997. Interface by J. Mackowiak, matching algorithm L. Schomaker)



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 on 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. Fig. 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 (Fig. *7a*). 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 normalisation is applied locally, in a sliding window, and is then called a local-affine transform. As an example, shear can be normalised by estimating handwriting slant and normalising to a given standard slant. Alternatively, a linear system estimation can be performed to minimise the distance between a character template and the current input window. The resulting minimum distance is then used in the character classification.

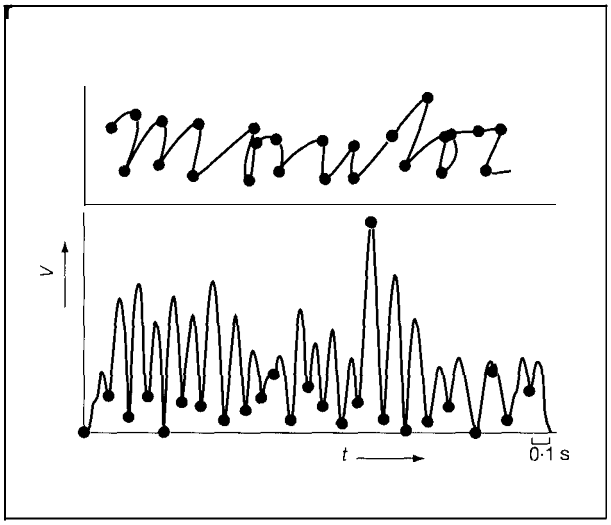


Fig. 6 A cursive-written word (monitor) and the time functon of the corresponding pen-tip velocity. Note that points of high curvature are characterised by a dip in the velocity. A trajectory between two velocity minima is called a 'stroke'

The second source of variation concerns *allographic variation*, or the number of character shapes used within the writer population for a given letter in the alphabet (Fig. 76). There are large shape differences between characters produced by different writers, especially when they are of different nationality or of different generations, or if they were taught different writing styles. This variation is the toughest problem in handwriting recognition and the main reason for the initiation of the UNIPEN"' project in which a database of 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 multilayer perceptrons or hidden-Markov recognisers. Such classifiers run the risk of generating average, but incomplete, representations of the total ensemble of allographic variants. Fig. 8 shows hierarchical clustering results for a set of 1800 randomly drawn characters: 600 letter g's, 600 letter f's and 600 letter k's. Without manual intervention, the system" has detected a family tree of letter shapes (allographs). The ultimate goal is to provide 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.

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 now includes 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, processing speed and quality (Fig. *7c*). For instance, speeding-up writing means that more force should be produced to maintain 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" or Fourier descriptors'". Since some forms of sloppiness occur frequently and may be the same for several writers, large 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')[14]. The

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 a analogue axial-force transducer
z	height of pen tip w.r.t. the table plane
ϕ_x, ϕ_y	angles of pen w.r.t. the table plane
Switching	by thresholding of pen force p or based on additional button(s) on the pen

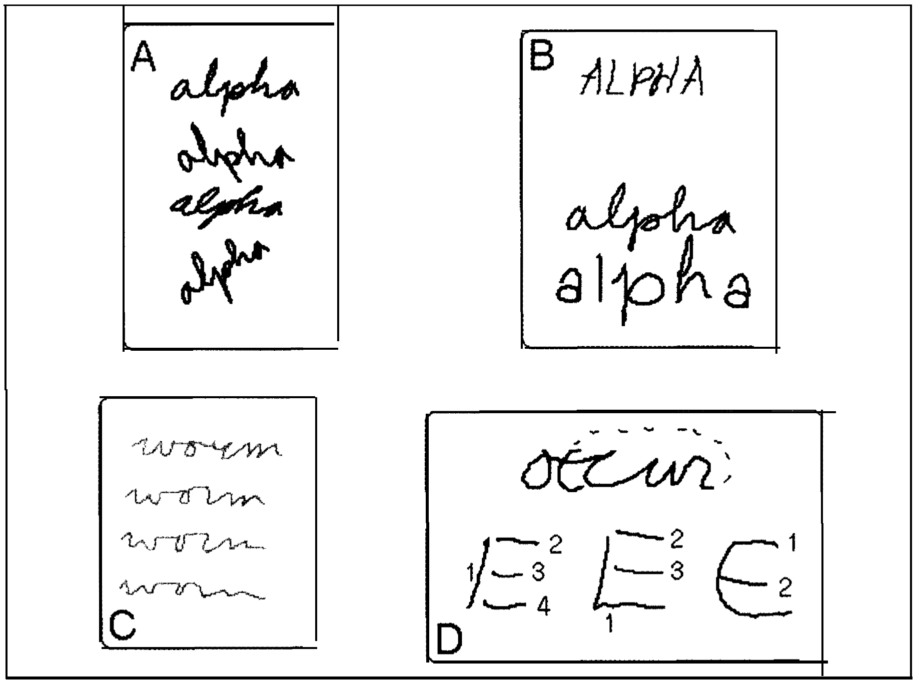


Fig. 7 Four basic sources of variation and variability in handwriting: (a) affine transforms; (b) allographic variation; (c) neuro-biomechanical variability; (d) sequencing variability. A robust recognition algorithm needs to solve the problems in all four areas

variability problems which are meant here fall in the shaping category; problems in chaining are categorised in the next paragraph.

The fourth source of variation, *sequence variability*, refers to the variable order in which handwriting may be produced (Fig. 7*d*). *Post* hoc editing — crossing of t's and dotting i's and j's — is typical. Also, spelling errors, due to limitations in the writer's linguistic knowledge, cause a problem. 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 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, human writers and readers often identify the nearest lexical match where an algorithm produces a *reject* response at best. Another example is the block capital *E*. If the *E* is produced in four strokes, each of which can be started at two ends, this results in $2^4 \times 4! = 384$ 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 recognisers of Chinese (Hanzi), Japanese (Kanji, Katakana, Hiragana) and Korean (Hangul) scripts.

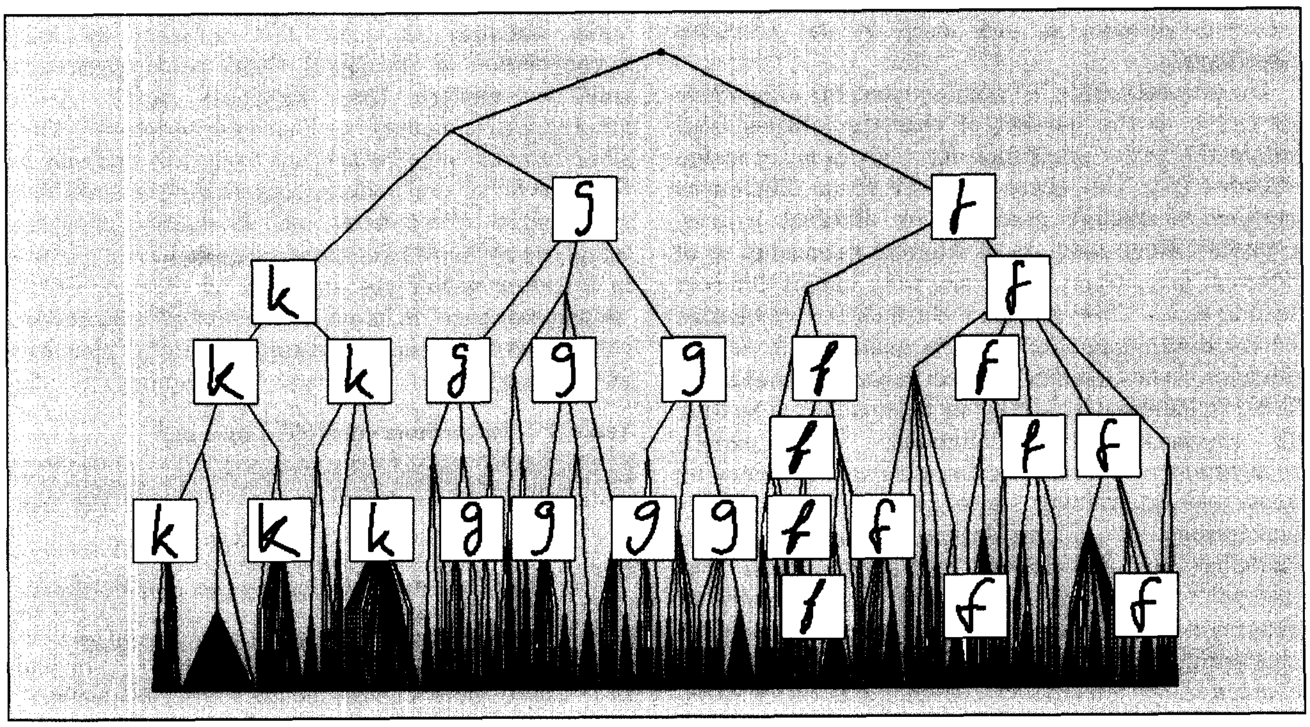


Fig. 8 A 'family tree' of character shapes for the letters k, g and f, as obtained by a new variant of hierarchical clustering developed at NICI[11]. The top node represents all 1800 input characters, whereas the nodes on the bottom line are the individual character samples (members). An allograph within a rectangle is the simple average $(\hat{x}, \hat{y})$ of all members

Input categories using handwriting

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

Table 4: Categories of handwriting shapes

Style	Description
I	digits
II	block capitals
III	isolated handprinted characters
IV	run-on handprint/mixed-cursive words
V	fully connected cursive words
VI	punctuation
VII	gestures, mark-up 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 recognisers is likely to be much higher than the performance of a word-oriented recogniser which includes digits as a possible input class. It should be noted that users do not make a distinction such as that given in Table 4 and will not understand if they are not allowed to write digits into a dialogue box intended for the recognition of isolated words. Similarly, the mixed use of capitals and connected-cursive script is difficult to handle, and recognition must be achieved by integrating the results of multiple classifiers which look at the same imput pattern. Apart from this coarse distinction in input categories many more subtle style variations exist, as shown previously in Fig. 8. The design of algorithms for combining information from several classifiers plays a central role in current reasearch. The following list summarises current basic pattern recognition methods:

- rule-based methods using structural character features and decision trees
- artificial neural networks (ANN) such as the multi-layer perceptron and Kohonen self-organising feature maps
- conventional 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 recogniser of on-line handwriting

The NICI stroke-based recogniser of on-line handwriting'" was developed on the basis of knowledge of the handwriting production process. Because it acts as an inverse transform of the human motor product, yielding the intended word identity, it was called Virtual Handwriting System (VHS)*. 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 writers produce ballistic movements without too many hesitations or other accidents, especially in connected-cursive script. The VHS approach is not suited to children's handwriting or handwriting with tremor.

Recognition performance measures should be interpreted with extreme caution. The rates are seldom underestimated in the literature. Fig. 9 gives a distribution of recognition rates for an unseen group of writers using the stroke-based recogniser (1995 version). The top-word recognition rate of the word classifier is how often the system's best guess was indeed correct. No word-shape information or linguistic statistics were used. The recogniser simply performed a strict search for individual letters, and all letters had to be found. This meant that all fused letters and spelling errors would lead to a missed word. The size of the lexicon was 250 words and each

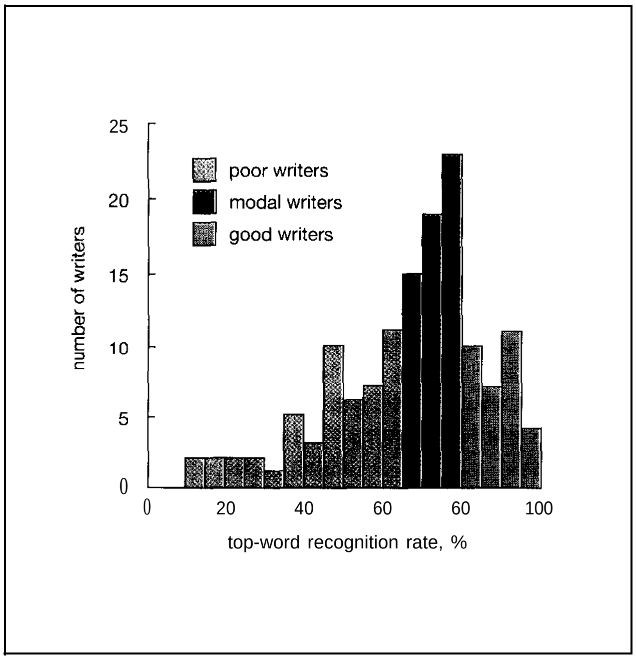


Fig. 9 A distribution of test-set recognition rates of the VHS handwriting recogniser. There are 140 writers in the test set. The modal group of writers will be able to enjoy top-word recognition rates between 65 and 80%. Good writers, i.e. those producing known shapes in a consistent way, will even achieve values between 80 and 100%. However, a large proportion of writers produce unknown shapes or write sloppily, yielding rates below 65%. More training data are clearly needed

*See http://hwr.nici.kun.nl/unipen/nici-stroke-based-recognizer/html

Lambert Schomaker received an MSc degree in psychophysiological psychology in 1983, and a Phi) degree on 'Simulation and recognition of handwriting movements' in 1991 at Nijmegen University, The Netherlands. Since 1988, he has been working in several European Esprit projects concerning the recognition of on-line, connected cursive script on the basis of knowledge of the handwriting movement process.

He was the project co-ordinator of a large European project on multimodality in multimedia interfaces (project MIAMI), and has engaged with collaborative research projects with several industrial companies. Current projects are in the areas of WWW information retrieval (PROFILE), image-based retrieval in large databases, and the benchmarking or handwriting recognition algorithms for pen computers and forensic handwriting analysis systems. Apart from research, he teaches cognitive ergonomics and neural-network-based modelling and pattern classification. He is the Chairman of the IAPR/TC11 committee for benchmarking of on-line handwriting recognisers (UNIPEN) and a member of the IAPR joint TC 11/TC5 committee on benchmarking and software.

Address: Nijmegen Institute for Cognition and Information, PO Box 9104, 6500 HE Nijmegen, The Netherlands.
Email: schomaker@nici.kun.nl

writer wrote 45 words. The average processing time on an HP-UX 9000/735 workstation was 215 ms per word. Note that when this system meets unseen writers, the recognition rate for a substantial number of them will be low. For example, some of the writers will write small 'all-caps' letters, claiming that that is their lower case handwriting. The VHS recogniser is only one of several methods we have tried over the last few years. Initially started as a pure connected-cursive recogniser, the approach gradually provided for incorporating mixed handwriting as well as isolated handprint.

6 Conclusion

In this paper, the problems of handwriting recognition and the development of applications for pen computers have been addressed. Despite the many problems, the use of the pen as an input device has survived, at least in small niches of the world of computer applications. Miniaturisation of computers necessitates a rethink of human-computer interaction design, in which the pen may still cover a substantial portion of the hutnan-computer channel bandwidth. Very often, a pen-based application will not require pattern recogniton at all, but will rely 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 regard the necessary improvement in handwriting recognition, current research topics are multiple-classifier integration, huge training sets (UNIPEN), and hidden-Markov modelling. Within our own research group, the current focus of attention is the development of allograph taxonomies using hierarchical clustering methods.

References

1 KAY, A.: 'User interface: a personal view. The art of human-computer interface design' (Addison Wesley, Reading, MA, 1990
2 FAIRHURST, M. C.: 'Signature verification revisited: promoting practical exploitation of biometric technology', *Electron. Commun. Eng.* J., December 1997, 9, (6). pp. 273-280
3 BELAID, A., and HATON, J.: 'A syntactic approach for handwritten mathematical formula recognition', *IEEE Trans. Pattern* Anal. Much. *Intell.*, January 1984, PAMI-6, (1), pp. 105–111
4 PAVLIDIS, J., and VAN WYK, C. J.: 'An automatic beautifier for drawings and illustrations', *Comput.* **Graph.**, July 1985, **19,** (3), pp. 225-234
5 FRANKISH, C., HULL, R., and MORGAN, P.: 'Recognition accuracy and user acceptance of pen interfaces'. Proc. ACM CHI'95, pp. 503-510 (Addison-Wesley, 1995)
6 VAN GELDEREN, T., JAMESON, A., and DUWAER, A. L.: 'Text recognition in pen-based computers: an empirical comparison of methods'. InterCHI'93 Conf. Proc., Amsterdam, 1993, pp. 87-88
7 GOLDBERG, D., and RICHARDSON, C.: 'Touch-typing with a stylus'. InterCHI'93 Conf. Proc., Amsterdam, 1993, pp. 80-87
8 SUEN, C. Y., BERTHOD, M., and MORI, S.: 'Automatic recognition of handprinted characters — the state of the art', *Proc.* IEEE, April 1980, **68,** (4), pp. 469-484
9 TAPPERT, C. C., SUEN, C. Y., and WAKAHARA, T.: 'The state of the art in on-line handwriting recognition', *IEEE Trans. Pattern Anal. Much. Intell.*, 1990, 12, pp. 787-808
10 GUYON, I., SCHOMAKER, L., PLAMONDON, R., LIBERMAN, R., and JANET, S.: 'Unipen project of on-line data exchange and recognizer benchmarks'. Proc. 12th Int. Conf. on Pattern Recognition, ICPR'94, Jerusalem, Israel, October 1994, pp. 29-33 (IAPR-IEEE)
11 VUURPIJL, L., and SCHOMAKER, L.: 'Finding structure in diversity: a hierarchical clustering method for the categorization of allographs in handwriting'. Proc. 1997 4th Int. Conf. on Document Analysis and Recognition, ICDAR'97, August 1997, pp. 387-393
12 KONDO, S.: 'A model of the handwriting process and stroke-structure of character-figures', in PLAMONDON, R., SUEN, C. Y., and SIMNER, M. L. (Eds.): 'Computer recognition and human production of handwriting' (World Scientific, 1989), pp. 103-118
13 IMPEDOVO, S., MARANGELLI, B., and FANELLI, A. M.: 'A Fourier descriptor set for recognizing nonstylized numerals', *IEEE Trans. Syst. Man Cybern.*, August 1978, SMC 8, (8), pp. 640–645
14 SCHOMAKER, L. R. B.: 'A neural-oscillator model of temporal pattern generation', *Hum. Mov. Sci.*, 1992, **11,** pp. 181-192
15 VARSHNEY, P. K.: 'Multisensor data fusion', *Electron. Commun. Eng. J.*, December 1997, **9,** (6), pp. 245-253
16 SCHOMAKER, L. R. B.: 'Using stroke- or character-based self-organizing maps in the recognition of on-line, connected cursive script', *Pattern Recognit.*, 1993, **26,** (3) pp. 443-450

This paper is a revised version of a keynote address presented at the 6th International Conference on Image Processing and its Applications held at Trinity College, Dublin, Ireland on 14-17 July 1997.