Proceedings ICPR 2000
to appear

Pen Computing: Challenges and Applications

Jayashree Subrahmonia, Thomas Zimmerman
IBM Thomas J Watson Research Center, USA
{jays@watson. ibm. corn, tzim@almaden. ibm. corn)

Abstract

Pen computing as afield broadly includes computers and applications in which a pen is the main input device. This field continues to draw a lot of attention from researchers because there are a number of applications where pen is the most convenient form of input. These include:

1. *Preparing a first draft of a document and concentrating on content creation.*
2. *A socially acceptable form of capturing information in meetings, that is quieter than typing and creates minimal visual barrier.*
3. *Applications that need privacy.*
4. *Entering letters in ideographic languages like Chinese- and Japanese; and non-letter entries like graphics. music and gestures.*
5. *Interaction with multi-modal systems.*

The advent of electronic tablets in the late 1950s precipitated considerable activity in the area of pen computing. This interest ebbed in the 1970 's, and was renewed in the 1980 's, primarily due to advances in pen hardware technology and improvement in user-interfaces and handwriting recognition algorithms. There are still however, a number of challenges that need to be addressed before pen computing can address the needs listed above to a acceptable level of user satisfaction.

In this paper, a overview of three aspects of pen computing are presented: pen input hardware, handwriting recognition and pen computer applications.

1. Introduction to Pen Input Hardware

The function of the pen input device is to convert pen tip position into X.Y coordinates at a sufficient temporal and spatial resolution for handwriting recognition and visual presentation. Pen tip contact with paper also needs to be sensed to establish when ink is deposited on paper. A pen input systems consists of a combination of pen. paper. and pad. To digitize handwriting, at least one of these elements must contain electronics.

1.1. Magnetic Tracking Technology

Figure 1 illustrates magnetic tracking. Sequentially energized coils embedded in the pad couple a magnetic field into a pen tank circuit (coil and capacitor). Neighboring coils pick up the magnetic field from the pen, and their relative strength determines pen location [Wacom]. The magnetic field can also be generated in the pen. requiring a battery that increases pen weight and thickness [Mutoh].

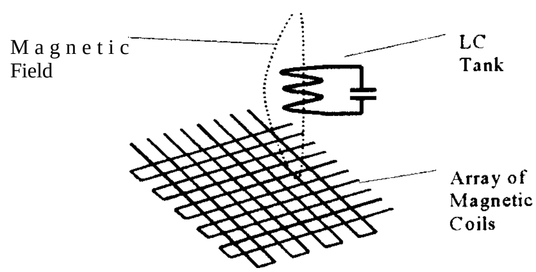

Figure 1 Magnetic Tracking

1.2. Electric Tracking Technology

Figure 2 illustrates how the conductive properties of a hand and normal pen can be used for tracking [Zimmerman]. A transmitter electrode in the pad couples a small displacement current through the paper to the hand. down through the pen. and back through the paper to an array of receiver electrodes. Pen location is calculated as the “center of mass” of the received signal strengths.

1.3. Ultrasonic Tracking Technology

Ultrasonic tracking is based on the relatively slow speed of sound in air (-330 m/sec). Referring to Figure 3. a pen generates a burst of acoustic energy. Electronics in the pad measure the time of arrival to two stationary

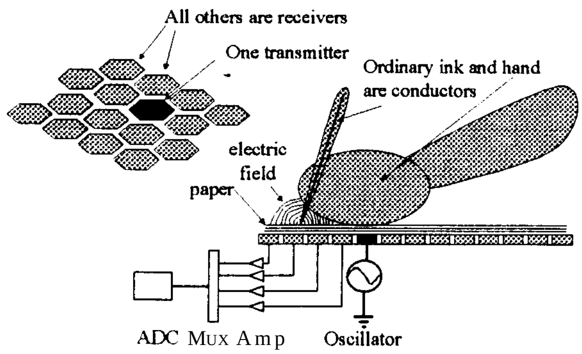

Figure 2 Electric Tracking

ultrasonic receivers [E-pen] [Mimio]. The ultrasonic transmission is either synchronized to the pad. typically with an infrared signal. or three microphones are required for unambiguous navigation[Yamaguchi].

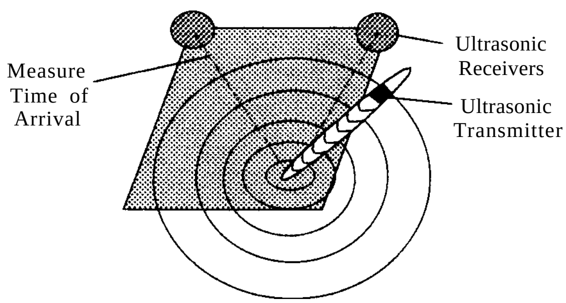

Figure 3 Ultrasonic Tracking

tracking (like a mouse) or absolute position tracking (like a touch screen). One method of relative optical tracking illustrated in Figure 4 shines coherent light from a laser onto paper and tracks the resulting speckle pattern. Another method mixes the scattered laser light with the source to create optical "beating" (similar to a Michelson interferometer), detected by a quadrant detector [Goulite] [Kinrot].

Bar codes printed over an entire page can provide absolute position waypoints for a tiny camera mounted in a pen [Anoto] [Lazzouni]. The bar codes can also encode page number. eliminating overwrites when a person forgets to tell the digitizer they have changed pages.

Another approach captures a sequence of small images of handwriting and mosaics them together to reconstruct the entire page [Nabeshima]. This method does not require special paper. but stops tracking when passed over stretches of blank (featureless) paper.

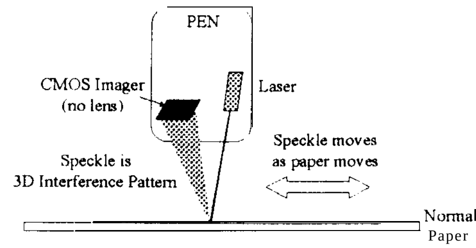

Figure 4 Optical Interferometry Tracking

1.5. Discussion of Input Hardware

Magnetic tracking is the widest deployed system due to high spatial resolution (> 1000 dpi). acceptable temporal resolution (>100 Hz). reliability and modest cost [CrossPad].

Magnetic and electric tracking require pad electronics and shielding. making them thicker and heavier then a conventional clipboard. Electric tracking uses a normal pen but has no direct way to measure pen tip contact. and must rely on less reliable pen trajectory analysis [Munich].

Ultrasonic tracking does not require pad electronics, making it lower cost and weight. Relative tracking can reach 256 dpi, but absolute spatial resolution is limited to about 50 dpi due to air currents that cause Doppler shifts.

(>2000 dpi) and temporal (>200 Hz) resolution. and can provide a self-contained pen that remembers everything written. Special bar code paper provides absolute position and page tracking. Optical methods based on CMOS technology lend themselves to low-power, low-cost. and highly integrated designs. These features suggest that optical tracking will play a significant role in future pen systems.

2. Handwriting Recognition

Pen can be used to record different kinds of information. Broadly. these can be classified into three groups:

1. Sequence of words from a language.
2. Drawing, where the information is pictorial
3. Non-alphanumeric symbols. which indicate a action to be performed.

Handwriting recognition is the problem of recognizing electronic ink (trace of the pen tip trajectory) belonging to any of the three groups listed above. In this paper, we will primarily concentrate on the first problem, i.e., that of recognizing sequence of words from a language. Recognition of pictorial information is typically done by segmenting it into a set of known shapes (squares, circles, lines etc.). Non-alphanumeric symbols are recognized using the same techniques that are used for word recognition. ([Schomaker], [Plamondon] and [Tappert] contain excellent reviews of different aspects of handwriting recognition)

Pen computers record handwriting information as a time ordered sequence of (x,y) points. The problem of recognizing writing in this case is referred to as *online* handwriting recognition, as opposed to ***offline*** handwriting recognition where handwriting information is captured as an image. Figure 5 shows a simplified data flow diagram for an online handwriting recognition system

The pen input device records the trajectory of the pen tip on the paper as a sequence of points sampled over time **(xt,yt)**. The set of points between a pen-down and next pen-up is called a *stroke.* The pressure of the pen tip on the paper may also be used during recognition.

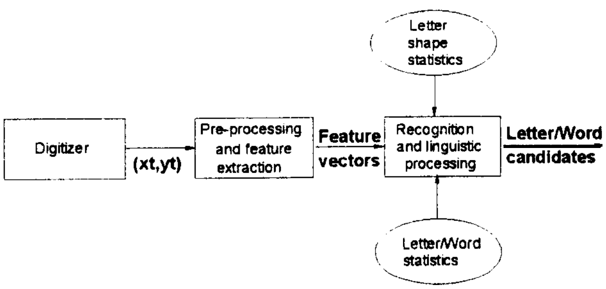

Figure 5: Online handwriting recognition system

Pre-processing of handwriting is done prior to recognition and typically involves noise reduction, normalization and segmentation. Feature vectors are then extracted from the pre-processed handwriting and are used in conjunction with letter and language models for recognition. The next three sections describe the functions performed by each module in Figure 5.

2.1. Pre-processing

Pre-processing receives raw position data from the digitizer and processes them into feature vectors by the following modules:

2.1.1. Noise reduction

Noise in the data typically originates from the limiting accuracy of the digitizing process. erratic hand motion and the inaccuracies of pen-down indication. A number of techniques are used for noise reduction. These include:

1. *Smoothing:* A common technique averages each point with its neighbors. Another well-known approach uses bezier curves to approximate the underlying ink trace.
2. *Filtering:* Eliminates duplicate data points and reduces the total number of points. The form of filtering depends on the recognition method. One filtering technique forces a minimum (or fixed) distance between consecutive points. When the writing is fast, however, the distance between successive points may far exceed the minimum. In this case interpolation can help obtain equally spaced points. Another filtering technique produces more points in regions of great curvature
3. *Wild point correction:* Replaces or eliminates occasional spurious points, typically caused by hardware problems.
4. *De-hooking:* Eliminates hooks that occur at the beginning and end of strokes, caused due to inaccuracies in pen down and pen up positions.

2. I. 2. Normalization

There is typically great variability in the size of letters. The goal of normalization is to reduce this variability. A number of algorithms are use for normalization:

1. *De-skewing:* Corrects slant. This can applied at the letter or word level.
2. *Baseline drift correction:* Orients the word relative to a baseline.
3. *Size normalization:* Adjusts the letter size to a standard size.
4. *Stroke length normalization:* Forces the number of points in a stroke to a specified number for easy alignment.

2.1.3. Segmentation

Segmentation is the process of breaking up the input handwritten data into smaller logical units. It can be done at various levels and is determined by the nature of the recognition algorithm:

1. *Stroke level segmentation:* Stroke level segmentation can be done trivially using pen-up/pen-down

features. This is used in recognizers like the Graffiti recognizer in Palm Pilots [Graffiti] where all letters are written using Single strokes.

2. *Letter level segmentation:* In printed writing, the user can explicitly indicate letter boundaries by writing in pre-defined boxes or using heuristic methods that look for spacing between adjacent strokes. In cursive writing, heuristics or statistics computed from training data are used. In most cases, all plausible segmentations are computed and the segmentation that results in the best recognition is used.
3. *Word level segmentation:* Word segmentation can be done temporally, spatially, or in combination. Word boundaries are determined temporally when the time between adjacent pen-down and pen-up events exceed a threshold. Word boundaries are determined spatially when the distance between the centroids of adjacent strokes exceeds a certain threshold.
4. *Line level segmentation:* Line segmentation uses temporal and spatial information to identify clusters of strokes that are contained on a line. If real-time recognition is not required, line segmentation is determined first, and word level segmentation is applied to strokes that belong to a line.
5. *Region level segmentation* If a user mixes pictorial information (drawings, tables etc.) with text, the page is first segmented by information type. Regions that are marked as text are then segmented into lines by the line level segmentation module.

2.3. Feature extraction

Features are typically extracted at a sub-letter level. The feature set varies greatly between recognizers. Some of the commonly use features are:

1. Shape descriptors: For example, ascender, descender, concave-down, concave-up, loop, cusp etc.
2. Symbolic representation of the singularities in a ink trace
3. Tangent and curvature features for a window of points along the ink trace
4. Writing speed

2.4. Recognition and Linguistic processing

The large variation in writing styles creates a great challenge to handwriting recognition systems. These variation come from:

1. Transformations -- scale, shear, rotation, slope, slant
2. Allographic variations – different styles of writing the same letter
3. Sloppiness – Users tend to write the same allograph with varying degrees of sloppiness.
4. Overwriting – Users tend to come back to a previously written word and correct it by either crossing out parts of the word and rewriting it. or inserting a sequence of letters in the middle of the word. This causes variation in the sequence of strokes used to form a word.

The goal of a recognition system is to convert written input into machine-readable text and graphics (e.g. ASCII text and graphics primitives). The task is broken down into a number of problems. Hence, the UNIPEN project [Guyon1] benchmarks a number of recognition scenarios. Examples of these include:

1. Isolated digit recognition
2. Isolated upper and lower case recognition
3. Isolated punctuation recognition
4. Isolated letter recognition (mixed case)
5. Isolated letter recognition in the context of words or text
6. Isolated printed word recognition
7. Isolated cursive or mixed-style recognition
8. Full page text recognition.

Online handwriting recognizers broadly use rule-based methods, statistical-based methods, or a combination of both. Most recognizers are letter-based, constructing words by identifying the letters that compose them. As the size of the lexicon grows, letter-based recognizers scale better than word-based recognizers. for the former requires less word level training data.

2.4.1. Rule based methods

Rule based methods use abstract descriptions of handwriting to recognize what was written. For example, a rule of recognizing the letter '**x**' might be 'two lines that cross over and are at approximately 45 and 135 degrees'. The problem with these methods is that it is not possible to design an exliaustive set of rules that model all possible ways of forming a letter. Variations of rule based methods that include fuzzy search seem to provide superior recognition performance, but are still not as good as the performance of statistical methods. They are however very useful in disambiguating between certain class pairs using very few parameters. One example, is the disambiguation between a '**v**' and '**u**' based on the presence or absence of a hook at the end.

2.4.2. Statistical Methods

There are three kinds of statistical methods that are used in handwriting recognition:

1. *Curve/feature matching..* Curve matching is a popular signal processing method where the unknown curve is compared against a set of prototype curves, and the closest match is selected. Matching is simplified by making the number of points in the curves the same. If the number of points differ, a popular comparison technique is *elastic matching* [Tappert]. Variations of this technique include matching feature vectors, such as stroke codes, instead of points for comparison [Tappert]. The problem with curve/feature matching approaches is that they are computationally intensive and impractical to use for large vocabulary cursive handwriting.

2. *Neural Network based approaches:* A well known neural network approach uses Multi-Layer Perceptron (MLP) training by back propogation [Bishop]. The training data is supervised. Input features are mapped onto the input units and the outputs units represent a class label (A class label in handwriting recognition might be a letter). These techniques work very well for small vocabulary, single letter recognition tasks. They however do not generalize well to cursive writing.

   A variation of this called convolutional Time-Delay Neural networks (TDNN) uses convolutional kernels for filtering [LeCun]. It uses fewer connections than MLP by defining a limited-size receptive field for units in the hidden layers. The input to a convolutional TDNN is a set of time-varying features and each hidden unit is a convolutional kemal. The output of the TDNN is a class label. The strength of this is its degree of invariance to spatial and temporal variations in the input. Also, the hidden units represent meaningful time and position independent features. The TDNN requires a substantially less number of parameters than MLP. The disadvantage with TDNNs is there inability to handle letters of varying lengths due to the fixed time window that they use to compute the input features.

   A variation of TDNN called Recurrent neural networks incorporating state recursion [Senior]. A part of the system state at time k is input into the system at time k+1. This allows the features to computed at a sub-letter level, which makes it less sensitive to letter length variations. Using recursion alleviates the loss of contextual information due to using sub-letter features. The neural network based approaches have small memory footprints making them attractive methods for small hand-held devices.

3. *Markov Model based approaches:* Hidden Markov Models (HMMs) have been used successfully for speech recognition and have recently been applied to the problem of handwriting recognition with great success [Jelenik] [Poritz] [Nathan]. Hidden Markov models are a type of Markov modeling where the sequencing of states modeling the formation of a letter is abstracted from observations.

   A typical HMM system models each letter by a series of states as shown in Figure 6. Topologies of the HMM can vary depending upon the application. They are all variations of the left-right model shown in Figure 6 with different sets of connections between states. Solid lines indicate transitions between states when a feature vector is observed, and dashed lines indicate transitions when no feature vectors are observed. Typically. a mixture of Gaussians is used to model feature vectors observed for each solid line transition. Transition probabilities associated with each solid and dashed line model the process of forming a letter. The number of states, parameters of the Gaussians, mixture coefficients that model the output distribution associated with each solid line and the transition probabilities define a HMM. Typically, each allograph is modeled by a HMM and the parameters are trained using the Expectation-Maximization algorithm from training data [Baum].

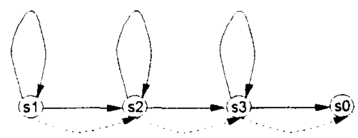

Figure 6: HMM Topology

2.5. Trainability

Out-of-the-box recognition is called writer independent recognition. The recognizer in this case is trained to optimize performance on a large sample of writing styles. There is however significant improvement in accuracy that can be obtained by customizing the recognizer for a single writing style [Subrahmonia]. Recognition is this case is called writer dependent recognition. The advantage of Markov-model based approaches over others is their ability to customize the models to a specific writing style. This is typically done by collecting handwriting data for a specific writing style and using it to optimize some or all of the parameters of the model.

2.6. Language Models

Language models contain linguistic information and used in conjunction with letter models for improved recognition performance. Language models can be applied at a number of levels ranging from phrases to single letters.

2.7. Complexity of a Recognition task

Figure 7 shows the three different factors that determine the complexity of the recognition task:

1. Content: The size of lexicon can vary from very small (for tasks like state name recognition) to open (for tasks like proper name recognition). In open vocabulary recognition, any sequence of letters is a plausible recognition result and this is the most difficult scenario for a recognizer.
2. Letter shapes: A tightly constrained system like Graffiti [Graffiti] forces users to write each letter in a pre-defined way. This reduces the variability considerably and hence makes the recognition task simpler. A completely unconstrained system on the other hand allows users to write in their natural handwriting style, which improves usability, but makes the recognition task extremely difficult.
3. Letter models: Gut-of-the box recognition uses a writer-independent system. This gives a good average performance across different writing styles. However, there is considerable improvement in recognition accuracy that can be obtained by customizing the letter models to a specific writing style. Recognition in this case is called writer-dependent recognition

Typically a writer-dependent, tightly constrained, small vocabulary system gives very good recognition results which the accuracy being in the upper 90's. As one moves farther away from the origin on any of the axes, accuracy degrades rapidly. The recognition task defines the position in this three-dimensional space that is appropriate to give the required recognition accuracy at the desired level of usability.

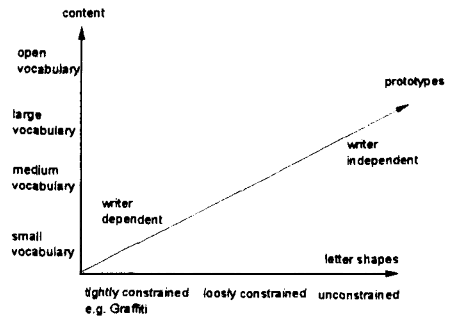

Figure 7: Complexity of a recognition task

3. Pen Computer Applications

Following is a list of some of the applications that currently use pen as a medium of input

1. *Form filling:* Many industries use forms for gathering information. Many forms are filled on the field where mobility is critical. Small hand-held devices (e.g. palm pilot, CrossPad) are used extensively in these cases for data entry and navigation for data collection.
2. *CAD applications:* Drawing is a task where the mouse is a extremely inconvenient mode of input. Electronic tablets by companies like Wacom [Wacom] are used extensively for sketching using pen as a mode of input.
3. *Note taking:* Note-taking in meetings is a application where the use of a keyboard is noisy and socially unacceptable. Paper and pen in this case are the acceptable mode of inputs. CrossPad [CrossPad] is a electronic pen platform that is used for note-taking applications using a pen-on-paper interface.
4. *Navigation:* Navigating through menus is greatly simplified by using a pen and pointing to items on the screen. Many emergency response applications use electronic pens and handheld devices to look at maps and navigate through them for identifying the source of the problem.
5. *Input of ideographic languages:* Use of the keyboard is extremely inconvenient for entering ideographic languages due to the complex sequence of key strokes needed to input a single letter. The pen is a much more convenient mode of input in these cases.

4. References

[Wacom] www.wacom.com/productinfo/features.html

[Mutoh] www.mutoh.com/pcoemfs.htm

[Zimmerman] T. Zimmerman and F. Hoffmann, IBM Research, patent pending, 1995

[E-pen] www e-pen.com

[Mimio] www.mimio.com

[Yamaguchi] N. Yamaguchi, H. Ishikawa, Y. Iwomoto, A. Iida "Ultrasonic Coordinate Input Apparatus", US Patent 5,637,839, June 10, 1997

[Goulite] www.goulite.com

[Kinrot] 0. Kinrot, U. Kinrot Laser Focus World "Interferometry: Encoder measures motion through interferometry". Volume 36 Issue 3. March 2000

[Anoto] www.anoto.com/download/anoto_desrip.pdf

[Lazzouni] M. Lazzouni, A. Kazeroonian, D. Gholizadeh, O.Ali, "Pen and paper information recording system". United States Patent Number 5,652,412, July 29, 1997

[Nabeshima] S. Nabeshima, S. Yamamoto, K. Agusa, T. Taguchi "Memo-Pen: A New Input Device" Chi'95 Proceedings Short Paper, 1995

[CrossPad] www.cross-pcg.com

[Munich] www.erc.caltech.edu/~perona/research/munich

[Graffiti] palmpilot.3com.com/products/input

[Tapper-t] C. C. Tappert, C. Y. Suen and T. Wakahara, *The State of the Art in On-line Handwriting Recognition,* IEEE PAMI, August 1990, pages 787-808

[Schomaker] L. Schomaker, *From handwriting analysis to pen-computer applications,* Electronics and Communication Engineering Journal, June 1998, pages 93-101

[Plamandon] R. Plamandon, D. P. Lopresti, L. R. B. Schomaker and R. Srihari, *Online Handwriting Recognition,* Wiley Encyclopedia of Electrical and Electronics Engg., Dept. of Electrical and Computer Engg.. University of Wisconsin-Madison

[LeCun] *Y.* LeCun and Y.Bengio, *Convolutional networks for images, speech, and time-series,* The Handbook of Brain Theory and Neural Networks, MIT Press. M. A. Arbib (ed), pp 255-258, 1995.

[Guyon] *I.* Guyon et. al., *Design of a neural network character recognizer for a touch terminal.* Pattern Recognition, pages 105-l 19, 1991.

[Jelenik] F. Jelenik, *Statistical methods for Speech recognition,* MIT Press. 1999

[Nathan] K. Nathan, H. S. M. Beigi, J. Subrahmonia, G. J. Clary and H. Mamyama. *Real-Time On-Line Unconstrained Handwriting Recognition Using Statistical Methods.* Proceedings of ICASSP, Detroit. 1995

[Subrahmonia] J. Subrahmonia. K. Nathan and M. Perrone. *Writer Dependent Recognition Of On-Line Unconstrained Handwriting.* Proceedings of ICASSP. Atlanta, GA, 1996,

[Guyon1] I. Guyon et. al. UNIPEN project on online data exchange and recognizer benchmarks, Proceedings of ICPR, 1994, pages 29 – 33.

[Bishop] C. M. Bishop, *Neural networks for Pattern Recognition,* Oxford: Clarendon Press, 1995.

[Poritz] A. B. Poritz, *Hidden Markov models: A guided tour,* Proceedings of ICASSP, 1988, pages 7 – 13.

[Baum] L. E. Baum. *An inequality and *associated maximization techniques in statistical estimation of probabilistic functions of Markvc processes.* Inequalities, *1972.* pages 1 – 8.

[Senior] A.W.Senior and A.J.Robinson *An Off-line Cursive Handwriting Recognition System.* IEEE PAMI Vol. 20, No. 3 March 1998 pages 309-321.