

# HANDWRITTEN SHORTHAND AND ITS FUTURE POTENTIAL FOR FAST MOBILE TEXT ENTRY

GRAHAM LEEDHAM

*School of Science and Technology, University of New England,  
Armidale, New South Wales 2351, Australia  
GLEedham@ieee.org*

YANG MA

*School of Computer Engineering, Nanyang Technological University,  
Singapore 639798  
MaYang@hotmail.com*

MICHAEL BLUMENSTEIN

*School of Information and Communication Technology, Griffith University,  
Gold Coast Campus, Queensland 4222, Australia  
M.Blumenstein@griffith.edu.au*

**Abstract.** Handwritten shorthand systems were devised to enable writers to record information on paper at fast speeds, ideally at the speed of speech. While they have been in existence for many years it is only since the 17<sup>th</sup> Century that widespread usage appeared. Several shorthand systems flourished until the introduction and widespread use of electronic recording and dictation machines in the 1970's. Since then, shorthand usage has been in rapid decline, but has not yet become a lost skill. Pitman shorthand has been shown to possess unique advantages as a means of fast text entry which is particularly applicable to hand-held devices in mobile environments. This paper presents progress and critical research issues for a Pitman/Renqun Shorthand Online Recognition System. Recognition and transcription experiments are reported which indicate a correct recognition and transcription rate of around 90% is currently possible.

*Keywords:* Shorthand, Hausdorff distance, Pen computing, Bayesian networks.

## 1. Introduction

The earliest identified examples of shorthand, or speed writing, date back about 2,500 years to the Greek and Roman civilisations and the Dynasties of Imperial China. The evolution of modern shorthand for English scripts began in the 16<sup>th</sup> Century. An early example is *Characterie; An Arte of Shorte, Swifte and Secrete Writing by Character* published by Timothy Bright in 1588 which introduced a system with 500 arbitrary signs resembling words; John Willis's *Art of Stenography* appeared in 1602, and Thomas Shelton's *Short Writing* appeared in 1626 (a system which, incidentally, was used at times by Sir Isaac Newton). Later, in 1720, John Byrom introduced the *New Universal Shorthand*, which was based on geometric shapes, followed soon afterwards, in 1786, by a similar system proposed by Samuel Taylor, which became widely used at the time. Taylor's system was subsequently superceded by Pitman shorthand which was introduced in 1837 by Isaac Pitman. Pitman shorthand, which is based on simple geometric symbols representing the consonant and vowel sounds of speech (Pitman (1980), Pitman (2003)), became the predominant shorthand system in the English speaking world (with the exception of the USA) right up to the present day. In the USA the Gregg system, introduced by John Gregg in 1888 and based on cursive symbols (Gregg et al. (1955)), first proposed by Franz Gabelsberger in 1834, has predominated.

By the 1960's shorthand was considered an essential part of secretarial training as well as being useful for journalists. Many colleges offered training courses and speed testing for intending secretaries. In the 1980's it was conservatively estimated that there were more than one million proficient users of Pitman shorthand and a similar number of Gregg shorthand writers. However, since the 1970's office technology has radically changed. The introduction of dictation machines reduced the importance of shorthand skills and subsequently, the widespread adoption of word processing applications on mobile computing devices such as PDAs and laptop computers has also reduced the usage of dictation devices. Only in specific areas such as court and news

reporting, or the Personal Assistant (human version) to Senior Executives are handwritten shorthand skills still in active use. While shorthand was originally intended as a means of fast note-taking, and the reader of the shorthand would usually be the original writer, it has been shown that Pitman shorthand has many features of a machinography – a machine compatible handwritten script, whilst Gregg shorthand is based on curves which are less compatible with current computer recognition techniques (Brooks, (1985)).

Pen-computing has been the unrealized dream since the 1950's when the earliest (non-portable) computers were beginning to appear. With the recent rapid advances in handheld devices such as PDAs, Palm Organisers and the Tablet PC, entering text information into such devices remains a serious bottleneck. Compared with other slow speed approaches such as the QWERTY keyboard (50-60 wpm), printed script (20-25 wpm) and cursive script (35 wpm with limited recognition performance), Pitman shorthand writers can readily achieve a recording speed of over 120 wpm. Currently, Pitman shorthand has been widely taught and used in 75 countries throughout the world. Over two decades ago, intensive research was carried out on the potential of Pitman shorthand as a means of rapid text entry to computer (Leedham (1984), Brooks (1985), Brooks & Newell (1985), Leedham & Downton (1986), Nair (1990), Qiao (1992), Nair & Leedham, (1992), Leedham & Qiao (1992), Qiao & Leedham (1993). At that time, there was no suitable pen-based device on which to exploit the approach. In the last ten years, a research group at Mysore University (Nagabhushan et al. (1999, 2000, 2002)) has been working on automation text production from offline Pitman notes. From their point of view, Pitman shorthand has advantages in being integrated into the present speech processing systems as it is the only universally accepted medium known for enabling real-time speech to text production.

With the recent widespread availability of inexpensive portable hand-held devices, it is now time to re-assess the potential of Pitman shorthand as a viable means of rapid handwritten text entry in mobile computer applications. Some promising algorithms have been arising recently for the recognition and transcription of Pitman shorthand (Ma et al. 2004, 2007; Htwe et al, 2004). Quite significantly, the Pitman shorthand structure is also applicable to Mandarin Chinese where the phonetic sounds of Chinese are mapped to the geometric strokes of the Pitman-style shorthand (Ma & Leedham, 2007) in a shorthand system called Renqun (Liao (1985)). Examples of Pitman and Renqun shorthand are shown in Figures 1 and 2. There has been some work reported for recognising and transcribing Renqun shorthand (Chen & Yu (1998), Chen et al. (2000), Chen & Qiao (1997)).

If the accurate recognition of Pitman shorthand is achievable, the techniques are equally applicable to Mandarin Chinese (with a different transcription system) and thus the recognition engine is potentially usable by two of the world's most widely spoken languages – English and Mandarin Chinese. The potential users of such a system would be anyone who needs to make notes or record verbatim speech in real-time as it is spoken. This would include news reporters, students and secretaries. It is estimated that about 750 million people speak English and 1000 million speak Mandarin. If only 1 in 10,000 or 0.01% of these people found such a device useful there would be over 1.5 million users.

In the remainder of the this paper the critical issues which need to be addressed in a mobile usable shorthand recognition system are addressed along with details and results of current recognition and transcription approaches and performance.

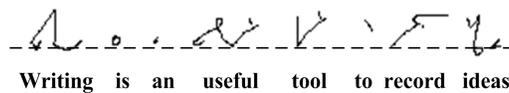


Figure 1. Example of an English sentence written in Pitman shorthand (shortforms are underlined)

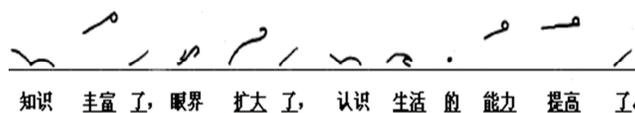


Figure 2. A Mandarin sentence written in Renqun shorthand (Shortforms are underlined)

## **2. Critical issues for a Pitman/Renqun shorthand recognition system**

There are four core issues that are critical in the future development and adoption of a mobile Pitman/Renqun Shorthand recognition system: (1) Algorithm Performance, (2) Software Engineering, (3) Processing Power and (4) Usability. Algorithm performance is mainly determined by outline recognition performance and transcription accuracy. Software Engineering and Usability determine the development of a commercially successful implementation. Besides algorithm performance, recognition speed is also a key factor in the commercialization process and this requires sufficient Processing Power. For successful commercialisation of a Pitman/Renqun Shorthand recognition system it should be able to run effectively on different device platforms including PDAs, smartphones, laptops, and tablet PCs. The computing capabilities of these platforms are increasing rapidly but the power-performance trade-off is becoming critical for mobile devices. This means that for small battery operated devices we are reaching a technological barrier which can only be surmounted through a quantum leap in battery and/or processor technology. We do not know when this will come so we should therefore concentrate attention on algorithms that need low computing requirement as well as producing good performance.

The major recognition and transcription issues that need to be resolved in Algorithm Performance are:

1. Pitman/Renqun shorthand contains little or no redundancy and is even an incomplete record of the words. That is, an outline may not be recognisable in isolation but has to be implied from the context and semantics of preceding and following outlines.
2. Pitman/Renqun shorthand provides little distinction between word outlines for totally different words and meaning, but having similar sounds (homophones) as in, for example, right/write/rite; hour/our, cite/sight/site.
3. Mapping an incomplete and inaccurate phonetic interpretation of a word to its true phonetic representation.
4. For Pitman shorthand: mapping a phonetic interpretation of words to their correctly spelt English words.
5. For Renqun shorthand: mapping a phonetic interpretation of words to their Pin Yin form, and then to their Chinese character.

## **3. A Pitman shorthand online recognition and transcription system**

### ***3.1 System Description***

A prototype system has been implemented on a Fujitsu Lifebook Tablet PC for the recognition and transcription of Pitman shorthand. All of the recognition and transcription software is implemented as an application-level program written in C and JAVA running on top of Windows. By implementing the system in this way using a high level language, standard operating system, and a general purpose Tablet PC, the effort required to port to other handheld devices with handwriting input is relatively low.

In the recognition engine, techniques in shape recognition, such as polygonal approximation, and Hausdorff Distance, are introduced into the recognition of Pitman notation. In this case, Pitman's handwriting can be viewed as a curved line rather than a handwritten word. The interpretation of segmented Pitman primitives into orthographic English words is achieved through the use of a Bayesian Network.

The structure of the system is shown in Figure 3. Firstly, shorthand outlines are written on the tablet surface and collected as a sequence of cartesian coordinates. Preprocessing performs noise reduction using a Gaussian filter, time normalization by applying equidistant resampling of the data, and size normalization in which the width of the bounding box is scaled to a standard width, and the height of the bounding box is adjusted to maintain the original aspect ratio. In the recognition engine, the outline is distinguished between a shortform and a vocalized outline depending on the number of pen-strokes, such that an outline with a single pen-stroke is assumed to be a shortform and an outline with two or more pen-strokes is assumed to be a vocalized outline. A template-matching approach, which directly produces a ranked list of English words is applied to classify shortforms. Therefore, the transcription system no longer needs to consider shortforms at word-level transcription unless a good match is not obtained in the shortform match, in which case it is interpreted as a

vowel-deficient vocalized outline. A two-stage approach is proposed to segment and classify vocalized outlines: firstly, in the segmentation stage, a non-parametric approach is used to select approximate dominant points, which indicate potential junctions between stroke features. Subsequently, the positions of dominant points are adjusted by polygonal approximation after special geometric features like circles, loops and hooks have been detected. In the classification stage, dummy dominant points are suppressed and merged. In the stroke reconstruction process, a heuristic merge algorithm is used to find all possible combination of segments. The interpretation of segmented Pitman primitives into orthographic English words is through the use of a Bayesian Network. Implementation of the network includes three subsections: (1) an outline model in which variations and dependencies between primitives are constructed, together with hidden nodes to represent missing vowel components; (2) an interference algorithm propagating the likelihood of primitive-nodes given a lexicon and training probabilities in an outline model; and (3) a unigram approach to select the best candidates for a given input outline.

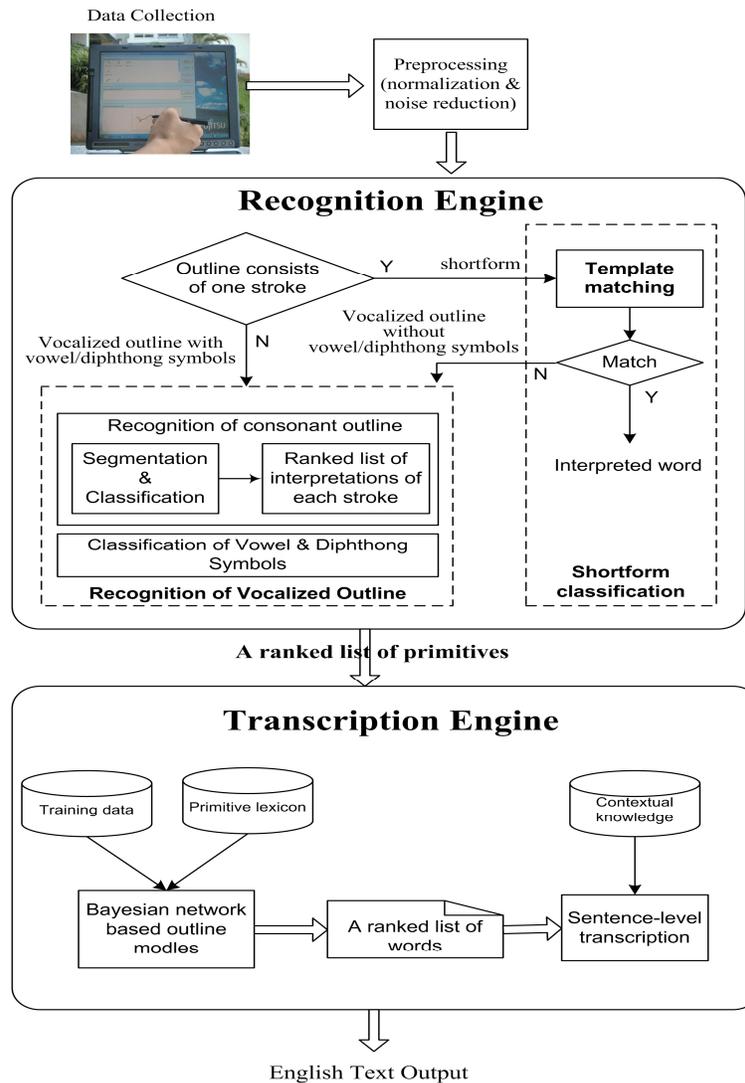


Figure 3. Overall structure of the Pitman shorthand recognition and transcription system

### 3.2 Classification of shortforms

Shortforms are classified using a template-based approach. For any two sample points  $m_i$  and  $n_j$  in the input outline and template, their dissimilarity is calculated on the basis of a polar coordinate system. The start point of

the pen-stroke is defined as the origin. Each point in the outline is defined by its distance and its angle referred to the origin.  $d(m_i, n_j) = d_l(m_i, n_j) + d_\theta(m_i, n_j)$ , where  $d_l(m_i, n_j) = |r_{m_i} - r_{n_j}|$  ( $r_{m_i}$  and  $r_{n_j}$  are distances from the origin to point  $m_i$  and point  $n_j$  respectively) and  $d_\theta(m_i, n_j)$  is the angular distance which is calculated by the approach used by Chen et al. (2002).

The Hausdorff Distance has been used frequently for shape matching and was used in this work to evaluate the degree of mismatch between the input shortform outline and a set of templates. Here, a generalization of the Hausdorff Distance which is insensitive to a single ‘outlying’ point of  $M$  or  $N$ , is used to measure the dissimilarity between the input outline and the template:

$$h(M, N) = \sum_{m \in M, n \in N} \min \|m_i - n_j\| / p$$

where  $M$  is the input outline,  $N$  is the template,  $p$  is the number of points in  $M$ .

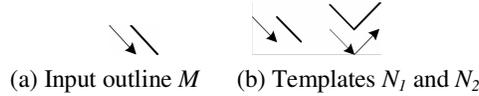


Fig. 5: Shortform shapes with similar Hausdorff distance

As shown in Figure 5, for the outline  $M$ , its Hausdorff distance to template  $N_1$  and  $N_2$  are equal to zero. To avoid such an ambiguity,  $h(N, M)$  is added to calculate the dissimilarity from the template to the input outline. The finalized Hausdorff distance includes two parts:

$$H = h(M, N) + h(N, M) = \frac{1}{P_m} \sum_{m_i \in M} \min_{n_j \in N, |m_i - n_j| < \delta_1} |d(m_i, n_j)| + \frac{1}{P_n} \sum_{n_j \in N} \min_{m_i \in M, |n_j - m_i| < \delta_1} |d(m_i, n_j)|$$

For point  $m_i$  in the input outline, point  $n_j$  in the template which satisfies  $|m_i - n_j| < \delta_1$  could be selected for detailed comparison. Parameters  $\delta_1$  and  $\delta_2$  define the allowed local warping. Large  $\delta_1$  and  $\delta_2$  may lead to possible redundancy; while small  $\delta_1$  and  $\delta_2$  impose constraints on the writing.

The template which has the minimal Hausdorff distance to the input outline is selected as the final classification result.

### 3.3 Recognizing the consonant outline

#### (1) Segmentation

The purpose of the segmentation is to produce a sequence of segmented consonant strokes which are easily classified in the next stage. Below, the three steps of this stage are described:

##### a. Dominant point detection

The segmentation approach is based on ‘the region of support’ to detect dominant points. Teh & Chin (1989) pointed out that determining the region of support is the main issue in dominant point detection. Incorrectly chosen regions of support may cause the measures of significance to be computed over inappropriate neighborhoods which may subsequently cause correct dominant points to be discarded. The region of support is symmetrical which is determined as follows:

For any point  $p_i$ ,  $l_{ik}$  is the length of the chord joining the points  $p_{i-k}$  and  $p_{i+k}$  as  $l_{ik} = |p_{i-k} p_{i+k}|$ ,  $d_{ik}$  is the perpendicular distance of the point  $p_i$  to the chord  $p_{i-k} p_{i+k}$ . Start with  $k=1$ . Compute  $l_{ik}$  and  $d_{ik}$  until

Rule 1:  $l_{i,k} \geq l_{i,k+1}$  or

Rule 2:  $d_{ik} / l_{ik} \geq d_{i,k+1} / l_{i,k+1}$  for  $d_{ik} > 0$

$d_{ik} / l_{ik} \leq d_{i,k+1} / l_{i,k+1}$  for  $d_{ik} < 0$

Besides the above two rules, two newly developed rules are used to preserve hook points. They are

$$\text{Rule 3: } Q_{i-k,i+k} \geq Q_{i-k,i+k+n} \text{ or } Q_{i-k,i+k} \geq Q_{i-k-n,i+k}$$

$$\text{Rule 4: } P_{i,i-k,i+k} \geq P_{i,i-k-n,i+k} \text{ or } P_{i,i-k,i+k} \geq P_{i,i-k,i+k+n}$$

where  $Q_{i,j}$  is the length of the straight line from point  $i$  to point  $j$ ;  $P_{i,j,k}$  is the perpendicular distance from point  $i$  to the straight line between point  $j$  and point  $k$ .

The region of support of point  $p_i$  is the set of points which satisfy any of the above rules and are denoted as  $D(p_i) = \{p_{i-k}, \dots, p_{i-1}, p_i, p_{i+1}, \dots, p_{i+k}\}$ . For points lying in the same region of support, only one with the maximal curvature value is selected as a dominant point.

#### b. Circles, loops & hooks detection

In order to detect circles and loops, the input outline must firstly be separated into segments at points whose preceding and succeeding chain code values are different. Each segment has a constant chain code value. Then each segment is compared to its previous segments to determine whether there is any overlap between them. If the intersection point of two overlapping segments lies in both segments, it is assumed that a loop or circle has been detected.

The hook identification is triggered if the distance between two consecutive dominant points is smaller than a predefined threshold. Extension, indicated by the dashed lines in Figure 6, is made at a corner point A to find any intersection possibilities with the input outline. The direction of the extension is orthogonal to line AB. If an intersection point C is detected, it is assumed that a hook has been found.

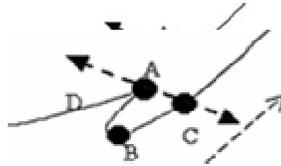


Figure 6. Hook identification

#### c. Dominant point adjustment by polygonal approximation

A split-and-merge approach based on polygonal approximation is introduced to adjust the positions of dominant points. Two polygonal approximation parameters (the ratio of the chord length over the straight length of the two consecutive dominant points; the area between the chord and the straight line) are firstly calculated. If any parameters are larger than experientially determined thresholds, a split algorithm is triggered. The point with the longest perpendicular distance to the straight line joining the two dominant points is chosen as a new dominant point. To avoid loss of information-rich points, merging is performed cautiously. Only collinear dominant points can be merged. The above steps are implemented recursively until polygonal approximation between any two consecutive dominant points satisfies the above criteria.

#### (2) Classification

A neural network and a stroke/non-stroke discriminator were constructed as the stroke/non-stroke and primitive category classifier in the above processes. And a heuristic search algorithm was applied to find the optimal path of the strokes.

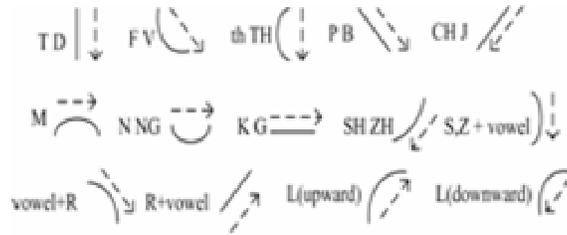


Figure 7: 14 types of basic consonant symbols and their associated consonant sounds

#### a. Configuration of the neural network

A 9-25-2 back-propagation neural network was used to classify whether each segment is a stroke or not. Two categories of patterns: positive patterns and negative patterns were used as training samples. There are 14 types of primitives in Pitman shorthand if stroke thickness and circles, loops and hooks are not considered as shown in Figure 7. The positive patterns are the 14 types of basic consonant strokes. The negative patterns are non-stroke patterns which are the possible combinations of basic consonant strokes. A 9-30-14 back-propagation neural network was used to determine the class to which the stroke belongs. 14 output dimensions denote the 14 types of strokes.

Nine features were extracted as the input vectors for the above two neural classifiers:

- the ratio of the straight length over the length of the chord from the starting point to the end point;
- the ratio of the width over the perimeter of the bounding box of the input outline;
- the ratio of the height over the perimeter of the bounding box of the input outline;
- the sine values of the direction of the straight line from the starting point to the end point;
- the cosine values of the direction of the straight line from the starting point to the end point;
- the maximal, average distance of all the points in the input outline to the straight line from the start point to the end point;
- the average distance of all the points in the input outline to the straight line from the start point to the end point;
- the percentage of negative distance points; and
- the percentage of positive distance points.

The above 9 features were separately normalized to the range 0 to 1.

#### b. Stroke reconstruction by heuristic merge

If  $n$  segments are left, at most  $2^{n-1}$  possible combinations are available. The search path will grow exponentially if nearby points are merged one by one to produce all possibilities. In our system, a heuristic merge algorithm is proposed to reduce the computational complexity of stroke reconstruction. For each dominant point  $p_i$ , a parameter named “importance” of a point ( $Im(p_i)$ ), which is the length from point  $p_i$  to the nearest point having a different chain code value from point  $p_i$ , is defined to improve the heuristic merge. If point  $p_i$  lies in a smooth region,  $Im(p_i)$  is large and  $p_i$  should be merged with high priority. Otherwise, if  $p_i$  lies in a sharp region,  $Im(p_i)$  is small and the possibility for point  $p_i$  as a correct segmentation point is high.

The following is the detailed search procedure:

First, the “importance” values for all dominant points are calculated. All these values are referred to as a set of merge thresholds, which is denoted as  $T = \{Im_1, \dots, Im_i, \dots, Im_{n+1}\}$ ,  $Im_i \leq Im_{i+1}$ . For any merge threshold  $Im_i$ , points satisfying the following three criteria should be merged one by one: (1) Points whose “importance” value is larger than  $Im_i$  have the possibility to be merged; (2) For each point, its merge priority is directly proportional to its “importance” value; (3) If a point is merged, the distance between its preceding point and its succeeding point should not exceed MaxSL.

Afterwards, the combination of segments generated from the merge is classified by the neural network and the discriminator. If the combination of segments does not contain any non-stroke segments, it is assumed that a path is found. Otherwise, the combination is regarded as invalid. For each available path, its “confidence” is equal to the summation of maximum interpretation of each segment divided by the number of segments. The

path with the largest “confidence” value is selected as the final one. For two paths with the same “confidence”, the one obtained at a smaller  $Im_i$  is regarded as more reliable. In this system, all available paths and their ranked list interpretation of each segment are ordered by their “confidence” and transferred to the transcription engine.

### 3.4 Transcription engine

The transcription engine consists of a collection of outline models where each model is constructed by concatenating segmented primitives with specified dependency from left to right in the writing order. The model is retained in a hierarchical structure such that the root node corresponds to an outline  $O$  relating to a particular word and each leaf node  $K_i$  ( $i = 1, 2, \dots, n$ ) corresponds to primitives comprised in the formation of the outline  $O$ . The task of the network is to return an outline  $O$  with a corresponding word token that maximises the conditional probability  $P(O | K_1, \dots, K_n)$ , where  $K_1, \dots, K_n$  are primitives classified by the recognition engine. Upon the arrival of new evidence, the model is retained according to three reasonable architecture assumptions. Our assumptions are based on the design concepts applied in signature verification and can be described as follows: *Unique node assumption*: In an outline model, if a particular primitive appears in almost every sample of an outline (e.g.  $K_3$  in Figure 8), the primitive can be considered as independent of other primitives and be linked directly to the root node. *Virtual node assumption*: In an outline model, if a primitive (e.g.,  $K_1$  in Figure 8) is directly dependent on a primitive  $K_2$  and there is an optional relationship between them (i.e., only one of them or none of them can be true at the same time), we can assume that there is a mechanism that controls the values of  $K_1$  and  $K_2$ , resulting in a virtual node  $U_1$  as shown in Figure 8. *Hidden node assumption*: If a primitive (e.g.,  $K_4$  in Figure 8) appears or disappears from time to time and the variation does not adhere to any rule (e.g., omissions of vowel components in an outline, according to the shorthand writers’ experience - the omitted locations are unpredictable) we can assume that there is a hidden mechanism that controls the value of  $K_4$ , resulting in a hidden node  $G_1$ . Hidden nodes can be considered as independent of other primitives and are linked directly to the root node.

#### 3.4.1 Inference Algorithm

The inference process of the Bayesian Network involves updating the probability of the primitives given some evidence and a priori probabilities (Xiao & Leedham (2002)). It is called the ‘belief’ of a primitive, denoted as  $BEL(P)$ . Among a variety of belief updating algorithms that support the Bayesian network, our work directly applies the message passing algorithm in which the belief of every node in the network is taken as the product of  $\pi$  and  $\lambda$  messages, where  $\pi$  is a message received from each of its parents (if any) and  $\lambda$  is a message received from each of its children (if any). Details of belief updating and propagation can be found in Pearl (1988).

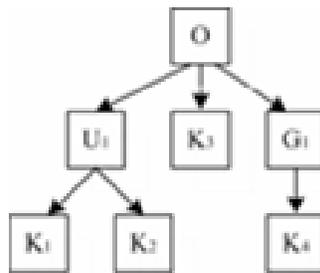


Figure 8. Bayesian network representation of an outline model

In our system, an outline model is constructed according to a lexicon initially and every node in the network is initialised with  $\pi$  and  $\lambda$  messages. Our message initialisation strategy is similar to the work of Xiao & Leedham (2002) in which the initialisation of  $\pi$  and  $\lambda$  varies depending the type of node in the network, but the estimations are different in our system. For each new word of the lexicon, a consonant-primitive is created as a

unique-node and a vowel-primitive is created as a hidden node in a corresponding outline model. The number of leaf nodes of the model at this stage is the same as the number of primitives contained in an outline. Each leaf node of the outline model is initialised at 1.0 and 1.0 for  $\pi$  and  $\lambda$  messages respectively, indicating that primitives of an outline appear in a lexicon are assumed to have 100% certainty. The root node is the topmost one in the network and does not have any parent; therefore its  $\pi$  message is set to 0.5 assuming there is an equal chance of taking a TRUE or FALSE value for this node. For each word in the training data, any primitives that deviate from the original pattern are added to a corresponding node, resulting in a virtual node with multiple children. A virtual node (or hidden node) is a judgemental node and its  $\pi$  message is set to 1.0, indicating that there is a true relationship to its parents and  $\lambda$  is set to  $P(\text{child\_nodes} | \text{observation})$ , indicating there is an optional relationship to its child depending on the observed value. The  $\lambda$  message of each leaf node at this stage is set by the normalized confidence score of the node obtained in the training process. The belief of a node is updated after the creation of every new node according to the formula:

$$BEL(x) = \beta \lambda(x) \pi(x)$$

where  $\beta$  is a normalization factor,  $\lambda(x)$  is a combined message received from all the children of node  $X$  and  $\pi(x)$  is a combined message received from all the parents of node  $X$ . Depending on the type of a node,  $\lambda(x)$  is calculated differently. If it is a root node,  $\lambda(x)$  can be defined by the formula derived by Pearl (1988):  $\lambda(x) = \prod_j (\lambda Y_j(X))$  where  $\lambda Y_j(X)$  is a message that node  $X$  received from its child node  $Y_j$ .

### 3.4.2 Model Selection

In Bayesian Network-based word recognition, an input outline is interpreted by finding a model that produces the highest a-posterior probability given a set of input primitives. We denote this process as “Model Selection”. A unigram-based heuristic is applied as the first level filter to produce a list of candidate models for a given outline.

“First and Last Segment Algorithm (F&L)” (Htwe et al., 2004) is the unigram approach used to filter outline models in our system. Human readers rely on the legibility of the first and last segments of a shorthand outline in the text transcription and the distribution of the two primitives can be defined as

$$P(x_i) = \frac{P(f_i)P(l_i)}{\sum_{i=1}^n P(f_i)P(l_i)}$$

where  $P(x_i)$  is the normalized probability of the first and last segment of an outline and  $\{f_1, f_2, \dots, f_n\}$  and  $\{l_1, l_2, \dots, l_n\}$  are the first and last primitives of  $n$  outline models. Models with the joint distribution above a threshold level are passed to the second level filter in which an a-posterior probability of each model is calculated based upon a set of input primitives and the ones with the greater posterior probabilities are chosen as the best candidate models for a given outline. If  $O_i$  is the  $i$ th outline model and  $K_1, K_2, \dots, K_n$  are input primitives classified by the neural classifier, the a-posterior probability of the outline  $O_i$  given the set of primitives can be defined as:

$$P(O_i | K_1, K_2, \dots, K_n) = P(O_i)P(K_1, K_2, \dots, K_n | O_i)$$

## 4. Evaluation

### 4.1 Data Collection

A sample sentence composed of words extracted from the most frequently used 5000 English words provided at <http://www.edict.com.hk/textanalyser/> was selected for the evaluation. The sentence consists of 20 shortforms and 29 vocalized outlines which include 58 consonant strokes, 11 hooks, 10 circles and 40 vowel and diphthong symbols. The average number of primitives per vocalized outline was 2.3.

An on-line Pitman Shorthand Recognition System (PSRS) as shown in Figure 9 implemented on a Fujitsu Lifebook T3010 Tablet PC was used for the experimentation. As a simulation system which demonstrates the interactive process between human and computer using Pitman shorthand, PSRS collects dynamic information (x,y coordinates, pen-down/pen-up indicators, time stamps etc.) of outlines written using the provided pen on the tablet screen and displays the output of recognition engine and transcription engine simultaneously. The user interface for this system was for evaluation only and would not be appropriate for a practical commercial system.

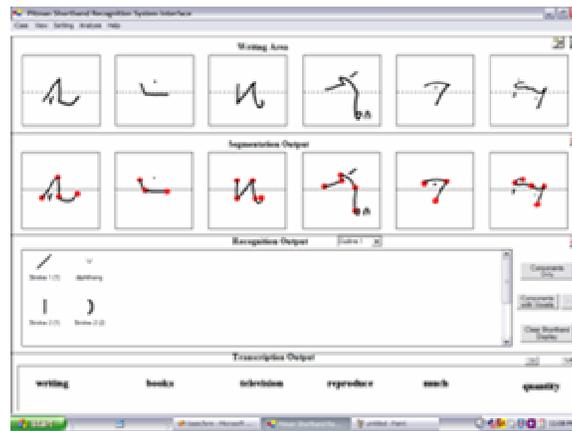
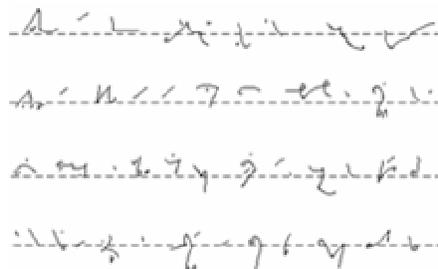


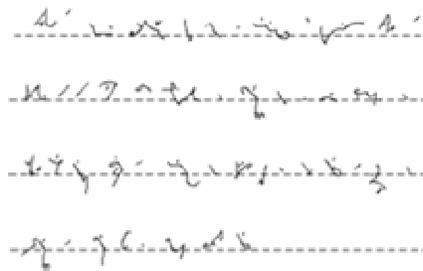
Figure 9: Interface of PSRS.

Each outline is written in a separate box (top row) and the recognition results showing dominant points (second row) and recognized pen strokes (third row) are indicated.

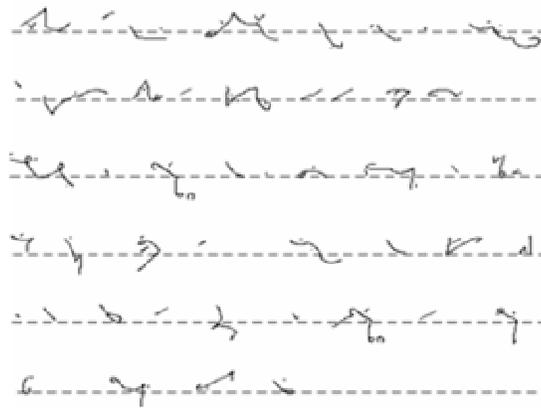
Three groups of three writers took part in the evaluation. The first group were proficient shorthand writers. The second group consisted of non-frequent users of Pitman shorthand. The third group were Pitman shorthand beginners. Before the experiments were carried out, the writers were given time to practice until they considered they had become proficient and familiar with writing on the screen of the Tablet PC using the special pen. This was typically readied after about 30 minutes' practice. A sample sheet with all standard outlines of the sentence was provided to each writer. Each writer was asked to write the sentence three times. Examples of the outlines collected are shown in Figure 10.



(a) Samples written by a proficient shorthand writer



(b) Samples written by non-frequent users of Pitman shorthand



(c) Samples written by a shorthand beginner

Figure 10: Examples of handwritten shorthand as written on the Tablet PC

#### 4.2 Recognition Accuracy

Table 1 shows the recognition rate of the shape recognition system. The correct classification rate of vowels and diphthongs was 97.5%. The overall recognition rate for vocalized outline was 77%. The position-sequence of vowels was not evaluated as it is common practice of stenographers to write them inaccurately and sometimes even omit vowels and diphthongs. The accuracy of shortform recognition was 89.4%. We observed an interesting phenomenon that the accuracy of beginners was higher than that of the other groups. The reason could be attributed to the beginners having developed fewer “bad” writing habits which disturb the recognition system and they paid more attention to the feedback of the recognition system. Samples written by shorthand beginners are closer to “text-book” quality. The size of outlines written by writers in the first and the second groups were relatively small (encouraged by Pitmans), which caused confusion between hooks and strokes due to larger relative size of hooks. In the experiments, shorthand beginners were willing to change their writing styles (eg. adjusting the size of hook in cases that were classified as a stroke /N/ or /NG/, writing boundaries with low angularity more clearly) from the feedback of the demonstration system to achieve higher recognition accuracy. Proficient writers were less likely to change the style of their writing and there were consistent errors (eg. failure in hook detection due to its small size, failure in detecting circles with direction reversal) in their writing. Given the accuracies for different stages and different signals, the average overall accuracy was 82.1%.

Table 2 shows the recognition rate of different processes in the segmentation stage. It can be seen that our algorithm performs well at identifying circles and loops. The average accuracy was 94.1%. The accuracy of dominant point detection (segmentation) and hook detection are 90.9% and 84.8% respectively. According to our analysis, segmentation errors mainly came from failure to detect dominant points at smooth boundaries. Sometimes the basic consonant strokes  $\smile$  which represent primitive /N/ or /NG/ were wrongly classified as a hook. This scenario happened particularly when the input outline was small so that the size of hook was relatively large causing some confusion between hook and stroke.

Table 3 shows the evaluation results of different processes in the classification stage. In the stroke construction procedure, the heuristic merge algorithm found correct primitive combinations (paths) for 98.1% of outlines. For outlines with a correct path, each segment was classified by the neural network to produce a ranked list of primitives. Among these ranked list primitives, 92.7% of the correct results are in the first position of the ranked list. This demonstrates the high classification ability of the neural network.

Table 1. Evaluation results of the recognition engine

Stage		Group 1	Group 2	Group 3	Average
Consonant outline	Segmentation Accuracy	81.6%	82.3%	86.2%	83.4%
	Classification accuracy	97.7%	98.0%	96.9%	97.5%
	Overall	75.6%	75.3%	80.2%	77.0%
Vowel & diphthong		92.2%	94.4%	97.2%	94.6%
Shortform		90.0%	88.9%	89.4%	89.4%
Overall accuracy		<b>81.5%</b>	<b>80.9%</b>	<b>84.0%</b>	<b>82.1%</b>

Table 2. Evaluation results in the segmentation stage

Accuracy	Group 1	Group 2	Group 3	Total
Dominant point detection	89.8%	92.1%	90.8%	90.9%
Hook detection	82.8%	83.8%	87.9%	84.8%
Circle & loop detection	90.0%	97.8%	94.4%	94.1%

Table 3. Evaluation results in the classification stage

Classification accuracy	Writer 1	Writer 2	Writer 3	Total
Stroke reconstruction accuracy	98.1%	99.0%	97.2%	98.1%
Primitives in the first place of ranked list	93.1%	93.2%	91.9%	92.7%

### **4.3 Transcription Accuracy**

In the statistical analysis we used a list of the 5000 most frequently used English words from the Brown Corpus. Based on this word list, we created two lexicons: one with a group of phonemes as the key and the other with a set of shorthand primitives as the key for each group of words, whereby the phonetic keys are extracted from the CMU phonetic dictionary and the primitive keys are automatically generated from phonemes of a word by applying the writing rules of Pitman shorthand to them.

Based on this study, 94% of the 5000 most frequently used English words have a unique representation. The maximum ambiguity is five potential words per index and an average ambiguity is three potential words per index. Therefore, a correct transcription accuracy of at least 94% can be estimated if there are no errors in the prior segmentation and classification of shorthand outlines.

The second experiment estimated the transcription performance in the presence of unclear thickness of a pen-stroke. This is the most common case experienced in the recognition of Pitman shorthand as most digitizers are unable to detect the thickness of a pen-stroke even though Pitman defines similar sounding consonants by the same strokes and differentiates between voiced and unvoiced sounds by thick and thin lines. It should be noted that regardless of the input technology, shorthand writers do not make a clear distinction between thick and thin strokes. According to this experiment, the ambiguity of the 5000 word phonetic lexicon increases by about 4% if there is no distinction between voiced and unvoiced consonants and, the transcription accuracy at best can therefore be expected to be 90%.

The third experiment attempted to predict the transcription performance in the presence of ambiguous vowel notations. This is an important consideration in the recognition of Pitman shorthand since vowels are frequently omitted in writing a Pitman outline and omitted positions vary from writers' experience or individual inclination. If a solution to the unpredicted omission of vowels in an outline is by the exclusion of vowels from the lexicon and matching without vowel components, the new version of the lexicon has about 71% unique indices.

The best transcription rate achieved by the Bayesian network approach was 92.86% with a 7.14% error rate due to inconsistent writing i.e. outlines which are legible to human readers, but are not consistent with the writing rules of Pitman shorthand.

### **4.4 Implementation Experience**

Most of the existing algorithms were first developed on a general desktop computer with a pen tablet. They were readily portable to the Tablet PC platform which has three distinct advantages: it has an inherent active digitizer, it has a large hard drive for permanent storage, and it is able to provide an identical development and debugging environment. These three factors allowed us to use the same development and debugging tools that were used on a PC. Familiarity with tools led to a much easier and quicker porting process to the platform of Tablet PC, in which the only effort was optimizing the pre-processing algorithms since the input interface was changed from a digitising tablet to the combined display/digitiser of the Tablet PC.

The recognition of consonant outlines, which are the most difficult task in Pitman/Remington shorthand recognition, had made a great improvement by introducing the techniques of a neural network. This algorithm performed well for circle & loop detection and classification. The average accuracies for circles and loops was 94.2% and 97.5% respectively. Further details of this algorithm can be found in Ma & Leedham (2007). Further details of the transcription algorithm can be found in Htwe et al. (2004).

Evaluation of the demonstration system shows that algorithm performance is still the central topic for future research, and it is also the most important factor in facilitating the commercialisation process.

## **5. Conclusion**

In this paper, a new solution to the recognition and transcription of Pitman shorthand is presented as a means of rapid text entry on PDA's and Tablet PC's. The system includes two major components: Recognition Engine and Transcription Engine. In the recognition engine, Shorthand outlines are differentiated between a vocalized outline and a shortform based on the number of pen-strokes used to create the outline. Shortforms, which represent 30% to 50% of all shorthands, are recognized separately from vocalized outlines using a Template

Matching Algorithm in which a ranked list of English words is produced. The vocalised consonant outline is segmented and classified by a two-stage approach. In the transcription engine, a Bayesian Network is used for the conversion from segmented primitives to the correctly spelt English text. In this approach, a lexicon is constructed as a collection of shorthand-outline models and an outline model is constructed as a collection of primitive nodes in a hierarchical structure. Each model retains the Bayesian network assumption and the belief (the likelihood) of each node is updated using various interference algorithms. Upon the arrival of evidence, models are firstly selected by the likelihood of the first and last primitive of an outline and secondly selected by the joint probabilities of all primitives of an outline.

Experiments using a sentence contains 20 shortforms and 29 vocalized outlines written by three groups of writers have shown that the average accuracy of the recognition engine and the transcription engine were 82.1% and 92.86% respectively. Observation of the three groups of writers with different experience in Pitman shorthand showed that shorthand beginners were most likely willing to adapt their writing habits to the recognition system. This fact also confirms our belief that if a stenographer regularly used an automatic recognition system he will be constrained to produce outlines with consistent high quality provided by the feedback of the transcription system. Future work needs to concentrate on enhancing the adaptability of both recognition engine and transcription engine.

This paper has also discussed the critical technological issues of commercializing Pitman/Renqun Shorthand recognition system with a demonstration system that illustrates the current state of Pitman/Renqun Shorthand recognition technique. These critical issues in the technological field come from two aspects: algorithm performance and software design. Given the current social and technological environment that Pitman/Renqun Shorthand is facing, it is essential to review the critical issues in order to take advantage of the potential of a Pitman/Renqun Shorthand recognition system.

Unless handwritten shorthand enters the technology age and is incorporated into mobile computing devices it will fall further into disuse and only used by the motivated few, as was the case 300-400 years ago. Of the surviving shorthand systems, still with a large user population, the Pitman/Renqun shorthand is the most machine compatible, or most closely related to a machinography. For shorthand transcription systems to become available and used, well designed, high performance, mobile products which prove themselves able to assist motivated people in their work by enabling real-time entry and transcription of English and Mandarin text into systems at high speed need to be developed in the next few years. This is now technically possible.

## 6. References

- Brooks C.P. & Newell A.F. (1985), Computer transcription of handwritten shorthand as an aid for the deaf—a feasibility study, *International Journal of Man Machine Studies*, Vol. 23, pp.45-60
- Brooks C.P. (1985), Computer transcription of Pitman's handwritten shorthand for the deaf, PhD Thesis, Department of Electronics, University of Southampton.
- Chen J., Leung M.K., Gao Y., (2002), Noisy logo recognition using line segment Hausdorff distance, *Pattern Recognition*, vol.36, pp. 943-955.
- Chen X. P., Yu, T.C. (1998) A study on recognition method of shortforms in Chinese handwritten shorthand, *Proceedings of the Fourth International Conference on Signal Processing*, 12-16 Oct., 1996, pp.1245 –1248.
- Chen, X. P., Yu, T.C. Dai R.W. (2000) Automatic Recognition of On-line Handwritten Chinese Shorthand, *Journal of Software*, vol.11, pp.1361-1367.
- Chen, X.P. Qiao, Y. Z. (1997) On-line Recognition of Renqun's Handwritten Shorthand, *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, 18-20 Aug. 1997, pp.941 –944.
- Dubuisson M.P. & Jain A.K. (1994), A modified Hausdorff distance for object matching, *Proceeding of the 12th international conference on Pattern Recognition*, Jerusalem, Israel, 9-13 Oct.1994, 1, pp. 566-568
- Gregg J. R., Leslie L. A., and Zoubek C. E. (1955), *Gregg shorthand manual simplified: Second Edition*. New York: McGraw-Hill, Inc., ISBN 0-07-024548-7
- Htwe S.M., Higgins C.A., Leedham C.G. & Ma Y. (2004), Post Processing of Handwriting Pitman's Shorthand using Unigram and Heuristic Approaches, *Lecture Notes in Computer Science 3163*, Springer-Verlag, pp. 332-336

- Leedham C.G. (1989), Pitman's handwritten shorthand: machine recognition and transcription, *Proc. of the 4th International Graphonomics Society Conference on the Development of Graphical Skills*, Trondheim, Norway, 24-26 July 1989
- Leedham C.G. (1984), Computer acquisition and recognition of Pitman's handwritten shorthand, PhD Thesis, Department of Electronics, University of Southampton.
- Leedham C.G. & Downton A.C. (1986), On-line recognition of Pitman's shorthand: an evaluation of potential, *Int. Journal of Man-Machine Studies*, Vol. 24, pp. 375-393.
- Leedham C.G., Qiao Y., (1992) High speed text input to computer using handwriting, *Instructional Science*, Vol.21, pp.209-221
- Liao R. Q., (1985) Rules of Renqun shorthand, publisher unknown, China
- Ma Y., Leedham C.G., Higgins C.A., & Htwe S.M. (2004), Segmentation and recognition of vocalized outlines in Pitman shorthand, *Proceedings of the 17th International Conference on Pattern Recognition*, Vol. I, ISBN 0-7695-2128-2, pp. 441-444, Cambridge, UK, 23-26 August 2004, pp. 441-444
- Ma Y., & Leedham C.G. (2007), On-line recognition of handwritten Renqun shorthand for fast mobile Chinese text entry, *Pattern Recognition Letters*, Volume 28, Issue 7, pp 873-883.
- Nagabhushan, P., Anami, B.S., Guru, D.S., (2002), Dictionary supported generation of English text from Pitman shorthand scripted phonetic text, proceedings of Language Engineering Conference, pp. 33- 42
- Nagabhushan P., Anami B.S., (1999), A knowledge based approach for composing English text from phonetic text documented through Pitman shorthand language", ICCS-99, New Delhi, pp 318-327.
- Nagabhushan P., Murali S., (2000), Linear Hough transformation for recognition of stroke primitives in Pitman shorthand text, National Conference on Recent trends in Advanced Computing, Tirunelveli, pp 126–128.
- Nair A. (1990), On-line acquisition and recognition of shortforms in Pitman's handwritten shorthand, MSc Thesis, Department of Electronic Systems Engineering, University of Essex.
- Nair A. & Leedham C.G. (1992), Evaluation of dynamic programming algorithms for the recognition of shortforms in Pitman's shorthand, *Pattern Recognition Letters*, Vol. 13, pp. 605-612.
- Pearl, J., (1988), Probabilistic reasoning in intelligent systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc., San Mateo, CA
- Pitman Isaac, (2003), *Course in Isaac Pitman Shorthand*, Kessington Publishing, ISBN 0766 161692.
- Pitman 2000 shorthand—First Dictionary, (1980) Pitman Publishing Ltd.
- Qiao Y. (1992), Computer recognition of consonant outlines in handwritten Pitman shorthand, PhD Thesis, Department of Electronic Systems Engineering, University of Essex.
- Qiao Y. & Leedham C.G. (1993), Segmentation and recognition of handwritten Pitman shorthand outlines using an interactive heuristic search, *Pattern Recognition*, Vol.26, No.3, pp.433-441.
- Teh, C.H. & Chin R.T. (1989), On the detection of dominant points on digital curves", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.11, No.8, pp.859-872.
- Xiao, X., Leedham, C.G., (2002), Signature verification using a modified Bayesian Network, *Pattern Recognition*. vol. 35, pp.983-995.