MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

OFF-LINE SHORTHAND RECOGNITION SYSTEM

R.Ethindra Kumar , A.Agarwal , Arun K. Pujari

Artificial Intelligence Lab. University of Hyderabad, Hyderabad 500134
INDIA

ABSTRACT

This work explores the possibility of fast man-machine communication through printed shorthand scripts. A structural approach is used to represent and recognise printed shorthand scripts wherein the primitive strokes and the vowels are separated in the preprocessing step . After a model-based recognition , the primitive strokes are again concatenated to identify the english word. The efficient recognition of the shapes, position and thickness of strokes and vowels are main features of this work.

INTRODUCTION

In applications such as verbatim reporting and office dictation, the ability to produce a simultaneous printed transcript or computer compatible textfile of spoken language is often desirable. Unfortunately, typists using the conventional *QWERTY* keyboard can not achieve even dictation speed of 120 wpm, while typical verbatim speed is 180 wpm. Automatic online recognition of speech offers an obvious solution to this problem but the recognition of connected speech of unlimited vocabulary requires many more years of directed research. An alternative means of obtaining real-time verbatim data entry is to use one of the *shorthand* systems [2] which have been developed specifically for verbatim recording. These systems record speech phonetically. Handwritten shorthand scripts of news-reporters (prepared at a remote site) require fast but off-line automatic conversion to equivalent English text.

The present study is concerned with automatic recognition of printed shorthand characters and in the process exploring the new technique of pattern recognition. The *Pitman* shorthand scripts has certain unique structural characteristics which is the most desired features for automatic recognition. This prompted the authors to explore the study of pattern recognition techniques for recognition of printed shorthand characters .

Unlike the longhand , Pitman shorthand notation records speech phonetically.The system divides the English language into 40 phonems, comprising of 24 consonants,12 vowels and 4 dipthongs. Each word is represented by separately written outline consisting of a combination of these phonetic symbols. The consonants of the word are recorded first as a connected string of strokes, hooks, loops and circles. Any vowels and dipthongs occuring in the word are written in the small marks around this consonant kernel in one of the three position relative to the associated related stroke. The consonant kernel is also written in one of the three position relative to the writing line. The thickness of a stroke distinguishes between voiced and unvoiced sound. The consonant strokes can be halved to indicate a following sound of *t* or *d* and it can be doubled to indicate a sound of *ter*, *der* or, *dher*. In standard printed script the length of the normal stroke is one half of an inch.

There has been some earlier attempt to recognise shorthand characters [1-2] but mostly the study involves on-line recognition. On-line recognition of shorthand characters has many disadvantages. Readers are requested to refer [3] for detail discussion on on-line *vs* offline recognition for handwriiten chatacters. These problems are highlighted further as the shorthand scripts are dependent on pen-pressure, length of the strokes etc. Our aim is to recognise complete handwritten shorthand text and we feel that for such a purpose off-line method is more practical and suitable. In the present study we concentrate on printed character as an initial step towards complete recognition.There is no earlier attempt in this direction so far.

THE PROPOSED SYSTEM

In this section we discuss the design of our system for automatic recognition of shorthand characters. The block-diagram of the proposed system is shown in Figure 1. A brief description of each module is given below.The printed text is scanned by Datacopy Image Acquisition System which results in 8 to 10 pixels width characters.

PREPROCESSOR

Thinning :
The thinning algorithm proposed by Zhang and Suen[4], with modification by Lu and Wang [5] results in a single pixel width 4_connected skeleton. A modified algorithm

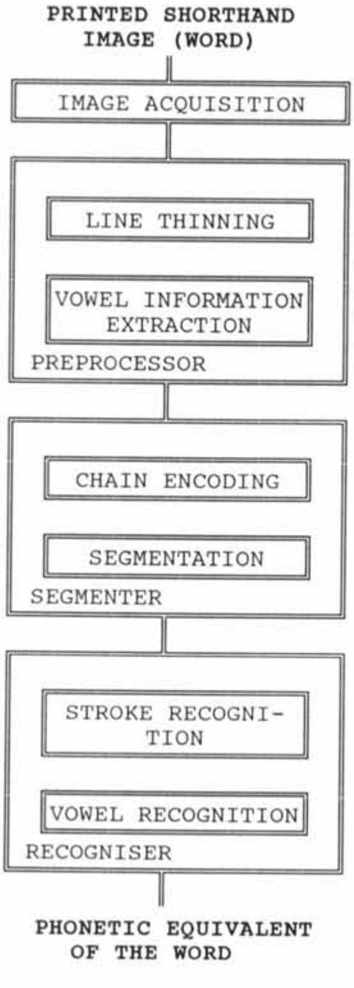

FIGURE 1 Block Diagram of the Prop[osed System

is used here so as to get a 8_connected skeletone using the following rules.
Rule 1 . An image point P_1 is deleted if it satisfies the following condition.

$$(P_2 \wedge P_4 \wedge P_8) \vee (P_4 \wedge P_6 \wedge P_8) \vee (P_2 \wedge P_6 \wedge P_8) \vee (P_2 \wedge P_4 \wedge P_8)$$

Rule 2: Similarly an image point is deleted if it satisfies the following,

$$(\bar{P}_9 \wedge P_4 \wedge P_8) \vee (\bar{P}_3 \wedge P_6 \wedge P_8) \vee (\bar{P}_7 \wedge P_2 \wedge P_4) \vee (\bar{P}_5 \wedge P_2 \wedge P_6)$$

Medial Axis Transformation :
The standard MAT is employed to determine the thickness of the strokes and identified as one of the two categories , namely normal strokes or, thick strokes.
Vowel Extraction :
A vowel (a dash or, a dot) is a disjoint connected component having smaller size than the standard strokes. Using a breadth_first_search the connected components are separated and taking the length into account the skeletal structure of the vowels are separated from the main kernel. The same process also helps in distinguishing a dash from a dot.

SEGMENTOR

Starting points:
Before extracting the chain code of the word , the starting point is identified by scanning the image in following six directions out of eight possible directions.
Scan 1 : Top to bottom, left to right columnwise,
Scan 2 : Left to right , top to bottom rowwise,
Scan 3 : Top to bottom ,right to left rowwise,
Scan 4 : Top to bottom, right to left columnwise,
Scan 5 : Bottom to top, left to right rowwise,
Scan 6 : Bottom to top, right to left rowwise.
It is observed that the starting point is one among the terminal points in these scan directions.
Chaincoding :
We extract the features during the process of chaincoding. This is due the fact that any pixel having three neighbours in a single pixel width 8_connected skeleton is most likely the joining (segmenting point) of more than one primitive strokes. Thus the chaincoding proceeds in a normal way till such a pixel is encountered . Among the two remaining neighbours of this pixel the one which represents the smoother change in

direction is taken as the next pixel in the chaincode. The chaincode is extracted until a terminal point or another 3_point junction is encountered. All these segmentation points and the terminal points are stored in an array with subscripts.

Segmentation:

The following rules are used for segmentation.

While computing the chaincode,certain pixels are already marked as the possible segmentation points.Such pixels are identified as the actual segmentation points based on the following rules.

(i) The strating point is a segmentation point.

(ii) The end point is a segmentation point if it is a terminal point. If it is a 3point junction then either one of the two 3point junctions is a segmentation point.

(iii) If the pixel has been marked as a possible segmenting point and the it is at least 35 pixels away in chaincode from the previous segmentation point.

(iv) If the distance is less than 35 pixels but more than 15 pixels then the point is a segmenting point with the resulting stroke being marked as a half stroke.

(v) If there is an abrupt change in direction with distance being more than 15 pixels are also marked as segmentation points.

(vi) If the current pixel is a starting point of a straight line then it is a segmentation point.

RECOGNISER

At this stage the the chaincode of the kernel together with the endpoints of the component strokes are available. The information regarding vowels, thickness are also separately available . The purpose of this module is to first identify the component basic strokes of the consonant_kernel and using the other information to form the equivalent word.

Primitive stroke recognition.

Various model primitives are stored as a chaincode and for each primitive model the count of eight chaincode directions are also determined. These counts for the primitive strokes are compared with those of the observed component strokes. Thus the cumbersome process of pixelwise matching is avoided and each feature vector is now of reduced dimension (one for each of eight direction) . The standard stroke is of 30 pixelwidth and by such a reduction the time for recognition is improved drastically. It is observed that this process gives very accurate results.

Vowel recognition :

We have already found whether a vowel is a dash/dot and thick/thin in the earlier module. Hence the problem is to decide to which component stroke the current vowel is associated and its position with respect to this stroke. There are six possible positions _ three on either side _ of a vowel to occur relative to the associated component stroke. If the vowel sound is spoken before the associated consonant sound then the vowel is written to the left and it appears to the right otherwise. Moreover the symbol (a dash or, a dot) represents different vowels depending on its position with respect to the stroke. This information is extracted from the processed image by finding the nearest consonant stroke to the vowel and its relative position. To find whether a vowel belongs to the current component stroke the distance between the vowel and the stroke is used. A vowel associated with a stroke would be closer to that stroke compared to the other component strokes in the same kernel In order to determine the position of the vowel, the stroke primitives are divided into three classes, (i) vertical (ii) horizontal (iii) slant strokes. In class (i) the vowel occurs either to the left or right and scanning the row of the vowel decides the relative position. In class (ii) , the vowel occurs above or below and hence scanning is done in the column of the vowel . In class (iii) , the vowel and the associated stroke have common row as well as common column . The position of the stroke nearest to the vowel determines its relative position.

RESULTS

In Figures 2-7 all features of the recognition process are illustrated. For the purpose of experiment we have chosen kernels with doubled strokes, halved strokes, circular joins, hooks, loops etc . The results show that all these categories are recognised correctly.

CONCLUSION

In this work we propose an off-line shorthand character recognition system. It is demonstrated that the system in the present form recognises a single printed vocalised outline with multiple vowels. The design work is being continued to successfully

FIGURE 2

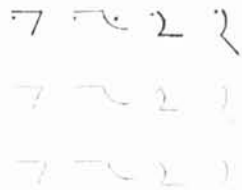

FIGURE 3

FIGURE 4

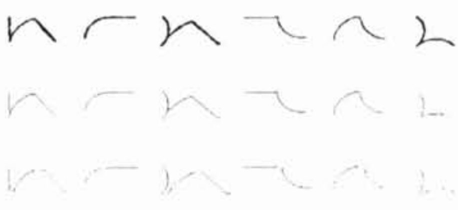

FIGURE 5

FIGURE 6

FIGURE 7

recognise a full text which includes shortforms and multiple outlines.

REFERENCES

[1]C.G.Leedham,A.C.Downton,C.P.Brooks and A.F.Newell, *Online acquisition of Pitman's shorthand as a means of rapid data entry,* Proc. 1st Intl. Conf. on Human-Computer Interaction, London, U.K. (1984).

[2] C.G.Leedham and A.C.Downton, *Automatic recognition and transcription of Pitman's handwritten shorthand - an approach to shortforms*, Pattern Recognition, 20(1987), pp 341-348.

[3] C.C.Tappert,C.Y.Suen and T.Wakahara, *The state of the art in on-line handwritten recognition*, IEEE-PAMI 12(1990), pp 787-808.

[4] T.Y.Zhang and C.Y.Suen, *A fast parallel algorithm for thinning digital patterns,* CACM 27(1984), pp 136-139.

[5] H.E.Lu and P.S.P.Wang , *An improved fast parallel thinning algorithm for digital patterns,* Proc. IEEE CVPR(85).

FIGURE 2 and 3 Strokes with characters written upwards and downwards.
FIGURE 4 and 5 Strokes with complex combination of primitive strokes.
FIGURE 6 Strokes of half length and double length.
FIGURE 7 Words with vowels and medial circles.