Indian Multi-Script Full Pin-code String Recognition for Postal Automation

U. Pal[1], R. K. Roy[1], K. Roy[2] and F. Kimura[3]
[1]*Computer Vision and Pattern Recognition Unit; Indian Statistical Institute, Kolkata-108, India*
[2]*Dept. of Comp. Sc., West Bengal State University, Barasat, India*
[3]*Graduate School of Engg., Mie University; 1577 Kurimamachiya-cho, TSU, 514-8507, Japan*
Email: umapada@isical.ac.in

Abstract

Under three-language formula, the destination address block of postal document of an Indian state is generally written in three languages: English, Hindi and the State official language. Because of inter-mixing of these scripts in postal address writings, it is very difficult to identify the script by which a pin-code is written. Also, because of the writing style of different individuals some of the digits in a pin-code string may touch with its neighboring digits. Accurate segmentation of such touching components into individual digits is a difficult task. To avoid such difficulties, in this paper we proposed a tri-lingual (English, Hindi and Bangla) 6-digit full pin-code string recognition. We obtained 99.01% reliability from our proposed system when error and rejection rates are 0.83% and 15.27%, respectively.

1. Introduction

Postal automation is a topic of research interest for more than two decades and many pieces of published article are available towards postal automation of non-Indian language documents [1,4,5]. At present postal sorting machines are available in several countries like USA, UK, Canada, Japan, France, Germany etc. There are only a few works on Indian postal system [3,7,8] and at present no postal automation machine is available for India. Indian pin-code (postal code) is a six-digit number and system development towards Indian postal automation is more difficult and challenging than that of other country because of its multi-lingual and multi-script behavior. In India there are 22 official languages and 11 scripts are used to write these languages. Although there are many languages in India, the destination address block of a postal document in an Indian state is generally written in three language: English, Hindi or the State official language. Thus Indian postal documents are tri-lingual in nature. To take care of such tri-lingual documents, in this paper we proposed a tri-lingual (English, Hindi and Bangla) pin-code string recognition. To the best of our knowledge this is the first work on Indian multi-script pin-code string recognition. Bangla is the official language of West Bengal State of India and Hindi is the Indian national language. We computed different statistics from a database of 7500 postal documents collected from West Bengal state of India. For detail statistics see [8]. From the statistical analysis we found that 12.37%, 76.32% and 10.21% postal documents are written in Bangla, English and Devnagari script, respectively. Thus, development of multi-script postal documents is very useful. To get the idea about the shape of the digits considered in our experiment, a printed digit set of these three scripts are shown in Fig.1.

Fig.1. Examples of printed digits. (a) English (b) Corresponding Bangla and (c) Devnagari digit.

There are two approaches for the OCR of multi-script nature: (1) identify the script and then use appropriate OCR based on the script (2) develop a system capable to recognize samples of all the scripts. There are many pieces of published work on script identification [9]. The script identification from the address portion of a postal document is very complicated due to inter-mixing of scripts while writing postal address. It is found that some people write the destination address part of a postal document in two or more scripts. See Fig.2, where the destination address is written partly in Bangla/Devnagari and partly in English. Also, a single line of an Indian postal document may contain two or more scripts. So identification of script by which pin-code digits are written is very difficult task because of the inter-mixing of scripts. Also in the postal document, digits in a pin-code may touch. Two, three, four and five digit

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.171

touching strings are available in Indian pin-code. See Fig.3 where touching strings of different digits are shown. Accurate segmentation of such touching string is very difficult. To avoid such segmentation of touching string into individual digits as well as to avoid script identification problem, in this paper, we proposed multi-script pin-code recognition where six-digit pin-code string is considered as a word and the pin-code recognition problem is treated as lexicon free word recognition.

Fig.2. Two sample of multi-script document are shown.

The recognition approach proposed in this paper is as follows. Here, we are not giving the pin-code extraction method, for details of it see [8]. At first binarization of the pin-code is done and it is pre-segmented into possible components (individual digits or its parts). It is noted that when two or more digits touch each other in a pin-code they generate big cavity regions (spaces). For example see Fig.4 where cavity region is marked by gray. Because of this touching behavior, we use water reservoir based concept [6] for the pre-segmentation of pin-code into primitives. Each primitive ideally consists of a single digit or a sub-image of a single digit. Pre-segmented components of a pin-code are then merged into possible digits to get the best possible pin-code. In order to merge these primitive components into digits and to find optimum segmentation, dynamic programming (DP) is applied using total likelihood of digits as the objective function. To compute the likelihood of a digit, Modified Quadratic Discriminant Function (MQDF) based on directional feature is applied here.

2. Pin-code pre-segmentation and Feature extraction

To find optimal segmentation by the segmentation-recognition scheme using dynamic programming, we pre-segmented a pin-code into primitives. In digit pre-segmentation our aim was to get over-segmentation instead of under segmentation. As mentioned earlier when two or more digits sit side by side in a touching pin-code, they generate big cavity regions (spaces). Because of such touching nature we use water reservoir concept for digit pre-segmentation. For details about water reservoir concept see [6]. We consider top and bottom reservoirs for our purpose.

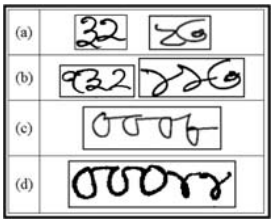

Fig.3. Example of touching Devnagari and Bangla digits. (a) Two digit (b) Three digit (c) Four digit (d) Five digit touching string.

To detect pre-segmentation point in a reservoir of a component, we first find two *candidate border points* of the reservoir. Computation of candidate border points is done as follows. From the reservoir base-line we consider the portion of the reservoir up to height $4*R_L$ and we note two border points of the reservoir at this height. (By base-line we mean a line, passing through the deepest point of a reservoir and parallel to its water flow level. R_L is stroke width and its computation is discussed later.) These two border points are considered as candidate border points. For illustration see Fig.4 where two candidate border points are shown and marked as A and B. Let the coordinates of these two candidate border points A and B be $(x_l\ y_l)$ and $(x_r\ y_r)$, respectively. Here y is the vertical axis and considered as row and x is the horizontal axis and considered as column. We scan the component vertically for each of the columns between x_l and x_r and find the number of crossing (black run) in each of the columns. The column from which we get minimum number of crossing is considered for pre-segmentation. If we get two or more such columns with minimum number of crossing then the column having minimum stroke width is considered for segmentation. Note that when two components touch, generally the stroke near the touching portion is thinner than the neighboring part. Based on this property, we decided to use the column having minimum stroke width for segmentation. If the height of a reservoir is less than $4*R_L$ then we consider the portion of the entire reservoir for candidate border points detection.

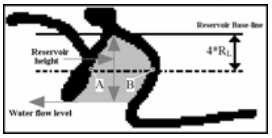

Fig.4. Candidate border point detection. Here two candidate border points A and B of a bottom reservoir are shown by small circles.

The pre-segmentation columns, obtained from a Bangla pin-code shown in Fig.5(a), are marked by small vertical lines in Fig.5(b). Please note that all the reservoirs are not considered for pre-segmentation. Those reservoirs having height greater than $1.5*R_L$ are considered for segmentation. Because of the structural shape of some digits, some small reservoirs may be

obtained where segmentation should not be done and we use this threshold to ignore the segmentations of such small reservoirs. In digit pre-segmentation our aim was to segment a pin-code into individual digits as much as possible avoiding much over segmentation. Note that we assumed there was no under segmentation in our pre-segmentation stage.

The stroke width (R_L) is calculated as follows. The image is scanned in row and column-wise and different run-lengths with their frequencies are computed. If a component has n different horizontal run-lengths r_1, r_2,..r_n with frequencies f_1, f_2 ...f_n, respectively, then $R_L=r_i$ where $f_i = \max(f_j)$, j = 1...n.

After detection of pre-segmentation columns from an input pin-code image, the image is split vertically at each pre-segmentation column and separated into horizontally non-overlapping zones. A connected component analysis is applied to the split image to detect the boxes enclosing each connected component. These boxes are usually disjoint and do not include parts of other connected components. Connected components in the split pin-code image and their enclosing boxes are shown in Fig.5(c). These boxes are numbered (from left to right) and these numbers are also shown in Fig.5(c). These connected components are regarded as primitive segments and each of which corresponds to a full digit or a part of a digit.

For our recognition purpose we have used two sets of feature [2]. For faster internal merging of pre-segmented primitives in dynamic programming we used 64-dimensional features. Histograms of direction chain code of the contour points of the components are used for this feature. Once internal merging is accomplished, higher dimensional feature (here we use 400-dimension) obtained from gradient information is used to get better accuracy. Feature extraction procedures are described below.

64 dimensional feature extraction. At first the bounding box is divided into 7 × 7 blocks (as shown in Fig.6c). In each of these blocks the direction chain code for each contour point is noted and frequency of direction codes is computed. Here we use chain code of four directions only [direction n0 (horizontal), n1 (45 degree slanted), n2 (vertical) and n3 (135 degree slanted)]. Thus, in each block, we get an array of four integer values representing the frequencies of chain code in these four directions. These frequencies are used as feature. Histogram of the values of these four direction codes in each block of a Bangla digit is shown in Fig.6(d). Thus, for 7 × 7 blocks we get 7 × 7 × 4 =196 features. To reduce the feature dimension, after the histogram calculation in 7 × 7 blocks, the blocks are down sampled into 4 × 4 blocks using a Gaussian filter [2]. As a result we have 64 (4 × 4 × 4) dimensional features for recognition. Histogram of these values of all the four directions obtained after down sampling is shown in Fig.6(e). The feature vector is normalized by dividing each component by the digit height to make it size independent.

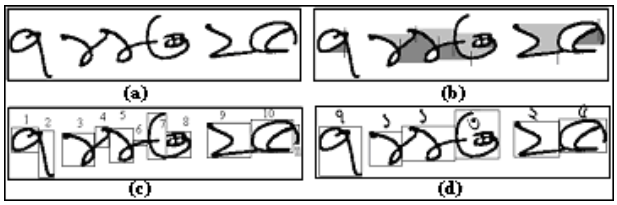

Fig.5. Example of digit segmentation from a Bangla pin-code. (a) An input pin-code. (b) Top and bottom reservoirs (marked by gray). Small vertical lines define pre-segmentation columns. (c) Individual segmented primitives are marked by disjoint boxes and numbered in English. (d) Optimum digit segmentation. Segmented digits and their respective printed Bangla digit are shown.

One critical point in segmentation-recognition techniques using dynamic programming is the speed of feature extraction, because the correct segmentation points have to be determined in optimization process with respect to the total likelihood of the resultant digits. The use of the cumulative orientation histogram enables one to realize high-speed feature extraction. Border following for feature extraction and orientation labeling are performed only once to an input pin-code image, and the orientation feature vector of a rectangular region including one or more boxes is extracted by a small number of arithmetic operations for high-speed feature extraction [2].

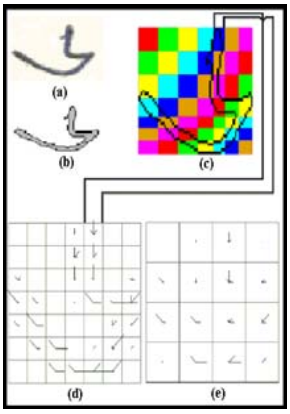

Fig.6. Example of feature extraction (a) Bangla digit six. (b) its contour (c) 7 x 7 segmented blocks shown in the zoomed version of 6(b). (d) Block-wise chain code histogram of contour points. (e) Chain code histogram after down sampling into 4 x 4 blocks from 7 x 7 blocks.

400 dimensional feature extraction. The accuracy of the classifier employed in internal merging is restricted due to the requirement on the computation time. However, once the internal merging of pre-segmented primitives is accomplished, a higher dimensional feature set can be applied to each segmented digit to

obtain more accurate digit likelihood. We use 400-dimensional features for the purpose and it is computed in the similar way as for 64-dimensional feature vector. The number of blocks is initially 9 x 9 and down sampled to 5 x 5. The quantization level is 16 directions instead of 4 orientations. To obtain 16 directions, Gaussian filter and Roberts filter are applied to a digit image to obtain a gradient image. The arc tangent of the gradient is quantized into 16 directions and the strength of the gradient is accumulated in each direction and in each block. For details of 400-dimensional feature extraction see [1].

3. Segmentation-recognition using dynamic programming

The core of a dynamic programming algorithm is the module that takes an input pin-code image, number of digits (= 6, because Indian pin-code string contains 6 digits), and a list of the primitives from the input pin-code image and returns a value that indicates the confidence that the input pin-code image represents the six digit numeral string. The primitive segments of an input pin-code image are merged and matched against the most likely digits so that the average digit likelihood is maximized using dynamic programming. The number of the primitive segments is usually 1 to 2 times as many as the number of digits in the pin-code. In order to merge these primitive components into digits and find the optimum digit segmentation, dynamic programming is applied using the total likelihood of digits as the objective function [2]. The number of digits in a pin-code is utilized in the process of dynamic programming to incorporate contextual information. The likelihood of each digit is calculated using the following modified quadratic discriminant function [2].

$$g(X)=\{|X-\hat{M}|^2 - \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_i + h^2}[\phi_i^T(X-\hat{M})]^2\}/h^2 + \ln[h^{2(n-k)}\prod_{i=1}^{k}(\lambda_i + h^2)]$$

where X denotes the input feature vector, $\hat{M}$ denotes the sample mean vector for each digit class, and λ_i and ϕ_i denote the eigen values and eigen vectors of the sample covariance matrix. Values of constant h^2 and k are selected experimentally to achieve the best compromise between speed and accuracy. In the following experiments, k is set to 20 and h^2 to $3/8*\sigma^2$, where $\sigma^2$is the mean of eigen values $\lambda_{i's}$ over i and digit classes.

Given a feature vector, $g(X)$ is calculated for all digit classes to find the maximum likelihood and the digit class. Based on the digit likelihood, total likelihood of a pin-code is found in terms of the dynamic programming technique as discussed above.

4. Result and discussions

For the experiment of the pin-code recognition scheme proposed in this paper we collected a total of 16300 (2692 Bangla, 8184 English and 5424 Devnagari pin-code string) handwritten pin-code string samples. Number of total pin-code class was 300 in each of three scripts. Minimum (maximum) number of samples in a class was 6 (40). These pin-code samples are collected from handwritten address block of Indian postal documents as well as from some individuals using some specially designed forms. We have used 5-fold cross validation scheme for recognition result computation. Here database is divided into 5 subsets and testing is done on each subset using other four subsets for learning. The recognition rates for all the test subsets are averaged to calculate recognition accuracy.

For recognition result computation we used different measures and they are defined as follows: Recognition rate = (N_C*100) / N_T, Error rate = (N_E*100) / N_T, Rejection rate = (N_R*100) / N_T, Reliability = (N_C*100) / (N_E+N_C), Where N_C is the number of correctly classified pin-codes, N_E is the number of misclassified pin-codes, N_R is the number of rejected pin-codes and N_T is the total number of pin-codes tested by the classifier. Here N_T= (N_C +N_E +N_R).

Global recognition results: From the experiment we noted that the overall pin-code string recognition accuracy of the proposed multi-script scheme was 94.14%, when 0.15% rejection was considered. Also, from the experiment we noted that overall 96.26% (96.68%) accuracy was obtained when first two (three) top choices of the recognition results were considered. Detail results of 3 scripts of our multi-script pin-code system with different choices are shown in Table 1.

Table 1: Pin-code recognition results based on different top choices (when rejection is 0.15%)

Number of top choices	Recognition rate (%)		
	Bangla	Devnagari	English
Top 1 choice	92.24	93.32	95.27
Top 2 choices	94.73	96.02	96.92
Top 3 choices	95.47	96.53	97.18

Rejection versus error rate: From the experiment we also computed reliability of our system. We obtained 99.01% reliability from our proposed system when error and rejection rates are 0.83% and 15.27%,

respectively. Pin-code recognition reliability with different rejection rates is given in Table 2. Rejection is done based on: (i) optimal likelihood value of the best recognized pin-code, and (ii) difference of the optimal likelihood values of the best and the second-best recognized pin-codes.

Table 2: Error and reliability results of the proposed system with respect to different rejection rates.

Reliability (%)	Error rate (%)	Rejection rate (%)
95.29	3.85	2.33
97.08	2.76	5.37
98.11	1.73	8.82
99.01	0.83	15.27

Comparison of results: To the best of our knowledge this is the first work on Indian multi-script 6-digit full pin-code string recognition where we considered Bangla, Devnagari and English pin-code strings in our system. Since there is no multi-script 6-digit full pin-code recognition results in the literature, we cannot compare our results. There exists many pieces of work on isolated numeral recognition and to get an idea of comparative results of isolated numeral recognition result we compare some results as shown in Table 3.

Table 3: Comparative results.

Script	Approach	Data	Accuracy
Bangla	Wen et al. [3]	16000	95.05%
	Roy et al. [8]	14650	96.66%
	Proposed method	16128	98.10%
Devnagari	Bhattacharya et al. [11]	22535	92.83%
	Hanmandlu and Murthy [10]	Not known	95.00%
	Proposed method	23340	98.41%

Error analysis: We also classified different pin-code recognition errors in term of the error in number of digit. We noted that 4.10% of the pin-code errors are one-digit error. In 0.85% (0.36%) cases the pin-code errors are two-digit (three-digit) errors. To get the idea about the erroneous samples, some examples of erroneous pin-code are shown in Fig.7.

5. Conclusion

In this paper we proposed a system for Indian multi-script handwritten 6-digit full pin-code string recognition and the pin-code string recognition problem is treated as lexicon free word recognition. Although there are many work on isolated digit but there is no work in the literature dealing with multi-script Indian 6-digt full pin-code recognition. This is the first work of its kind. We obtained 99.01% reliability from our proposed system when error and rejection rates are 0.83% and 15.27%, respectively.

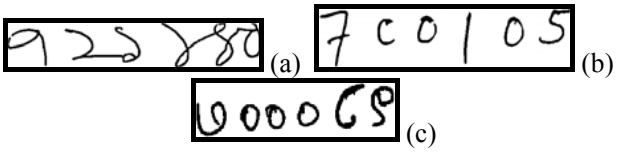
(a) (b) (c)

Fig.7. Examples of some miss-recognized pin-codes (a) Bangla pin-code ৭২১১৪০ is recognized as ৭২১১১০ (b) English pin-code 700105 is recognized as 760105 (c) Devnagari pin-code ७०००८९ is recognized as ७०००७९.

6. References

[1] F. Kimura, Y. Miyake and M. Sridhar, "Handwritten ZIP code recognition using Lexicon free word recognition algorithm", In Proc. 3rd ICDAR, pp. 906-910, 1995.

[2] F. Kimura, S. Tsuruoka, Y. Miyake and M. Shridhar, "A Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words", IEICE Trans. Inf. And Syst., Vol.E7-D, No.7, pp.785-793, 1994.

[3] Y. Wen, Y. Lu and P. Shi, "Handwritten Bangla digit recognition system and its application to postal automation", Pattern Recognition, vol.40, pp.99-107, 2007.

[4] L. Liu and M. Koga and H, Fujisawa, "Lexicon driven segmentation and recognition of handwritten character strings for Japanese address reading", IEEE Trans on PAMI, vol. 24, pp. 1425-1437, 2002.

[5] R. Plamondon and S. N. Srihari, "On-Line and off-line handwritten recognition: A comprehensive survey", IEEE Trans on PAMI, Vol.22, pp.62-84, 2000.

[6] U. Pal, A. Belaid and C. Choisy "Touching digit segmentation using water reservoir concept", PRL, vol. 24, pp. 261-272, 2003.

[7] U. Pal, K. Roy and F. Kimura, "A Lexicon driven method for unconstrained Bangla handwritten word recognition", In Proc. 10th IWFHR, pp. 601-606, 2006.

[8] K. Roy, "On the development of an optical character recognition system for Indian postal automation", Ph.D. Thesis, Jadavpur University, 2008.

[9] U. Pal,"Automatic Script Identification: A Survey", Vivek, vol.16, pp.26-35, 2006.

[10] M. Hanmandlu and O. V. Ramana Murthy, Fuzzy model based recognition of handwritten numerals, Pattern Recognition, vol.40, pp. 1840-1854, 2007.

[11] U. Bhattacharya et al., Neural combination of ANN and HMM for handwritten Devnagari numeral recognition, In Proc. 10th IWFHR, pp.613-618, 2006.