A Pragmatic Schema of Destination Address Interpretation for Use in Indian Postal System

P.Sri Rama Prasanna, S.Balaji, Thejavor Haralu Khezhie, C.Vasanthanayaki, S.Annadurai
Department Of Electronics and Communication Engineering,
Government College of Technology, Coimbatore – 641 013. India.
91 - 44 - 24336925
pavaniprasanna@yahoo.com

Abstract – Ever since character recognition originated, its potential use in the automation of mail sorting has been a vogue area of research. Despite over 50 years of research [2], this has not come to complete implementation even in highly voluminous postal systems like the Indian postal system. This paper proposes a pragmatic way of solving this historical myth by taking possible constrictions into consideration. Prior works of numerous labs [1] propose few solutions for this. Invariably all of these proposals are complicated by the fact that there are as many as 18 scheduled languages [3] in India. We propose a unified pragmatic solution for recognizing hand-written/machine-printed addresses in multiple languages. The Pragmatic Mail Processor (PMP) proposed by this paper is essentially a major advancement of Automatic Mail Processor (AMP) proposed by [1]. Unlike AMP, PMP adopts adaptive schemas for locating and processing the destination address block in a mail. Vertical and Horizontal smudging techniques have been successfully implemented for demarcating the individual lines and words in destination address. A robust field based context-sensitive retrospective holistic recognition system is used for address word interpretation. Context sensitive recognition is achieved by using a dynamic set of field specific vocabulary with the aid of a star search algorithm. The retrospective nature of the recognition system enables it to correct its previous flaws. Also, retrospective recognition provides a tight link between the adaptive segmentation module and the neural network based analytical recognition module. Robustness in recognition is achieved by longitudinal back traversing capability: the recognition system traverses forward normally and at times of inconsistency in results, it turns backward and tries to re-recognize the already recognized character. Multiple neural networks, one for each language, trained using the Fourier descriptors of chain codes are used in character recognition. After recognition, the destination address is represented in the form of a Delivery Point Code (DPC) [1], which is printed on the mail in the form of a barcode.

Index Terms – Smudging, Filtering, Contouring, Chain codes, Fourier descriptors, Neural Network, Star-search algorithm.

1. INTRODUCTION

Achieving total automation in mail sorting has been the dream of numerous character recognition labs for the past few decades. This has been achieved partially in few countries for machine printed destination addresses in one particular language. Practical implementation of this is tricky as most of the addresses are hand-written. In a country like India, the implementation is more complicated by as many as 18 scheduled languages [3]. There is no restriction that the address in a mail should be written in one particular language only. To accomplish a decently accurate recognition in such a scenario, the recognition system should be able to identify and interpret the characters of all these languages. The system should also be efficient enough to recognize words written in cursive style.

The line and word separation techniques proposed in the literature predominantly deal with scanning through the address image to identify blank pixel rows and columns. This is time consuming and is not suitable real-time processing. Traditional techniques [1] undertake a sequential processing style in recognition—every word is first segmented into different characters and recognition is done later. This style of operation is error prone especially given that the input is a hand-written word written in a cursive style. The proposed Pragmatic Mail Processor (PMP) adopts adaptive address block location and vertical/horizontal smudging algorithms for locating the destination address and for demarcating individual fields. A high accuracy in recognition is achieved by the neural network based analytical recognition engine and its encompassing field based context-sensitive retrospective holistic recognition system.

2. MODULES OF PRAGMATIC MAIL PROCESSOR

The PMP, which inherits its overall design from AMP [1], adds more functional value to it. The significant modules of PMP are Pre-processing, Redundancy correction, Horizontal and Vertical smudging, Recognition and DPC generation. A mail is first scanned using a binary scanner with high spatial resolution. A 1-bit image is sufficient in generating a image with its foreground distinctly different from its background. The binary image is de-noised with a median filter.

3. ADAPTIVE REDUNDANCY CORRECTION

Adaptive redundancy correction is employed for locating the destination address in the mail. The address location becomes simple in Inland letters and post-cards as they already have pre-defined address grids on them. For general envelopes, the address could be written anywhere on it and it should be first precisely located before further processing. The adaptive redundancy correction algorithm takes to its advantage the fact that the destination address is never written on the corners of an envelope.

Figure 1: Central address patch connected to redundant patch.

The de-noised binary image is low-pass filtered so as to accomplish sufficient blurring. Patches of information are formed when the low-pass filtered image in re-binarized. The unwanted information such as the stamp-markings, sender's address can be removed by traversing from the corners and chucking off the connected patches. The objective is to retain the central patch which essentially corresponds to the destination address. Such a system could end up removing the central patch also if the central path were to be connected to redundant corner patches (Fig 1). This necessitates the need for adaptive redundancy correction. In adaptive redundancy correction, the size of the spatial mask used for low-pass filtering the de-noised binary image is adaptively varied according to the output of this module. Essentially, there is a feed-back, which is used to vary the dimension of the spatial mask. Once the redundancies are removed, the actual destination address can be determined by plotting a rectangle of least area encompassing the central patch after leaving a tolerance level which is proportional to the dimensions of the spatial mask used for low-pass filtering.

4. HORIZONTAL AND VERTICAL SMUDGING

The horizontal and vertical smudging techniques are used for demarcating the lines and words of the destination address. Horizontal smudging refers to the process of blurring along horizontal coordinates and Vertical smudging is burring along vertical coordinates.

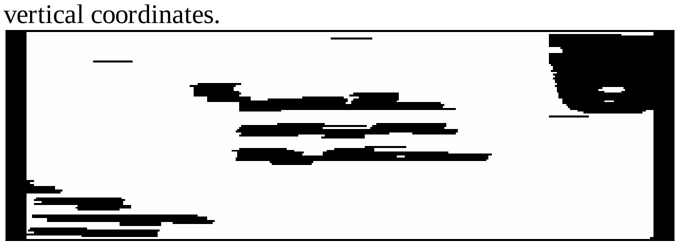

Figure 2: Horizontal smudging.

For three horizontally adjacent pixels p1, p2 and p3 with x-coordinates x1, x2 and x3, horizontal smudging with a 1x3 mask would result in the equation, $p2 = (p1 + p2 + p3)/3$. Similarly, vertical smudging can also be quantitatively described. Horizontal smudging of a binary image results in horizontally adjacent pixels of the image getting connected. The same phenomenon applies to vertical smudging where in vertically adjacent pixels get connected.

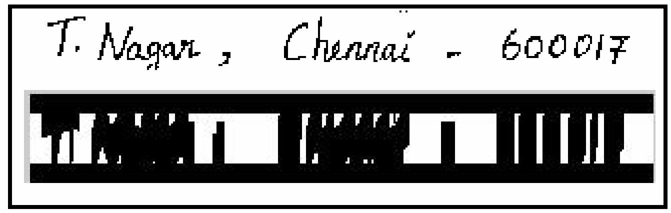

Figure 3: Vertical smudging of a line

The horizontal connectivity and vertical connectivity of adjacent pixels can be used for line and word separation efficiently. The destination address is first subjected to horizontal smudging. This would result in an image with streaks of patches connecting the address lines. By using the connected white spaces, the lines can be separated. The required lines are then extracted using the boundary coordinates obtained from horizontal smudging. These lines are then smudged vertically which would cause the word-pixels to be connected to form vertical streaks there by giving a clear indication of different words. The connected white spaces between adjacent words are used to demarcate between words. The words can now be extracted using the boundary coordinates obtained from vertical smudging.

5. RECOGNITION SYSTEM

In applications like address recognition, the accuracy in recognition can never be compromised. Most of the recognition techniques in literature confront the recognition of a word by segmenting them into individual characters and later identifying the characters. Such a system might work well with machine printed words. But, cursive hand-written words can never be recognized by such systems with the same level of accuracy. In cursive words, there is no clear demarcation between two characters. The curves joining two individual characters in a cursive word are called ligatures. Ligatures are specific to languages and there are also languages that do not permit the presence of ligatures. [1] proposes a way of separating ligatures by considering the slope of the characters besides them. Though such a system may work well with sufficient accuracy for a particular language, our programmatical experiments indicate that applying the same set of descriptor rules across different sets of languages would turn out to be erroneous.

The Pragmatic Mail Processor uses a field-based context sensitive retrospective holistic recognition system. The main components of the recognition system are a longitudinal shifter, a language identifier, a neural-network based analytical recognition engine, star-search data structures. As the recognition is field-based and context sensitive, any incorrect predictions of the recognition engine can be averted at the very early stage. For instance, there is a great deal of similarity between ‘S’ and ‘5’. If the field under consideration is a numeric field, like pin-code, the recognition system would do away with ‘S’ and report ‘5’ as the output. A dynamic vocabulary [2] is used to accomplish context sensitivity. The range of vocabulary used for recognition is field specific. The vocabulary used is called dynamic as it is used during the process of recognition. A static vocabulary is generally used after the recognition process to cross validate the results. The dynamic vocabulary used by PMP is organized in the form of star search data structures. As different types of fields have different vocabularies, PMP requires many star search data structures, one for each type. For instance, the pin-code field would have a data structure containing the list of all valid pin codes. Similarly, the city field would have a data structure containing the list of all valid cities. A typical star search data structure for English vocabulary is depicted in Figure 4.

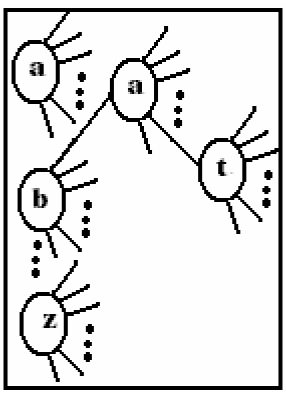

Figure 4: Star search data structure depicting the word “bat”.

The word to be recognized is first passed through a longitudinal shifter. Longitudinal shifter is a program that has the capacity to shift a word to its right or left through few pixel units. A physical model of a longitudinal shifter is shown in Figure. The amount of pixels through which a word is shifted is adaptively determined based on the length (of pixels) in the word and the approximate number of characters present in the word. The number of characters present in the word image can be approximately determined by subjecting the word image to vertical smudging and later performing thickness correction. Thickness correction basically refers to the process of removing patches with minimal thickness measures. This would result in connected white space patches between two characters. The shift length of the longitudinal shifter is proportional to the length of the word image and is inversely proportional to the number of characters in the word image.

A recognition unit is used in juxtaposition with the longitudinal shifter. The recognition unit encompasses a recognition engine. All the recognition engines used by PMP are neural network based analytical recognizers that accomplish recognition based on the Fourier descriptors of chain codes of a character [1]. The PMP uses 19 recognition engines of which one is a language identification engine and the others are language specific recognition engines. A recognition engine is basically a neural network that has been trained with the Fourier descriptors of all the characters of a particular language. The language identification engine is trained with all possible characters of all the 18 languages. The Fourier descriptors of a character are obtained by applying DFT on the chain codes of the outer contour of a character. The recognition engine in the recognition unit can be varied programmatically: for instance it could be a language identifying engine or it could be a recognition engine for one particular language. Before proceeding on with the recognition of an address, the language of the address has to be determined. Only after the language of the address is determined, the corresponding recognition engine for the language can be loaded into the recognition unit.

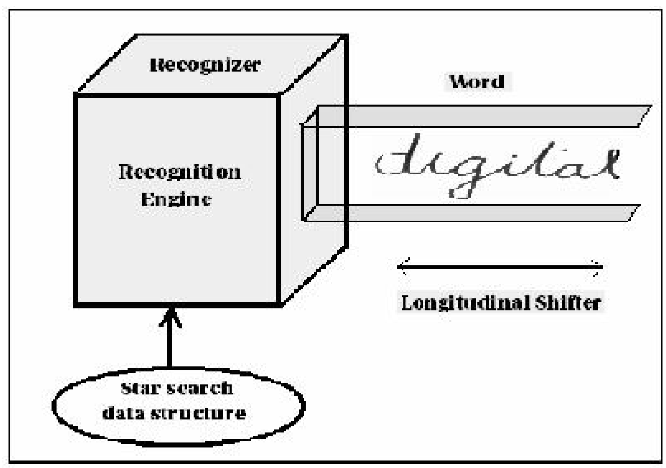

Figure 5: Recognition system

To accomplish this, first the language identification engine is loaded into the recognition unit and a word image is loaded into the longitudinal shifter. The language is determined based on the confidence level of the first ‘n’ characters. The actual process of retrospective recognition is explained in the next section. The language of the destination address is decided when high confidence measures for a particular language is obtained consistently for successive characters. Once the language is determined, the corresponding language recognition engine is loaded into the recognition unit.

The word to be recognized is initially shifted left by the longitudinal shifter so that the first part of the word enters the recognition unit. The recognition engine in the recognition unit recognizes the character under consideration and generates suitable matches with along with their confidence levels. The match with highest confidence level is initially assumed to be the recognized character. Note that at this point the recognition system just assumes the character and it does not decide the character. A character decision is made only if subsequent characters are confidently recognized. Once the recognition is done, the longitudinal shifter shifts the character image to the left and the recognizer tries to recognize the character under its view again. The outputs and their confidence levels are noted. This process proceeds until the confidence level curve reaches a peak and then slides down drastically. At this point, the character with the peak confidence level is decided and is removed from the word image.

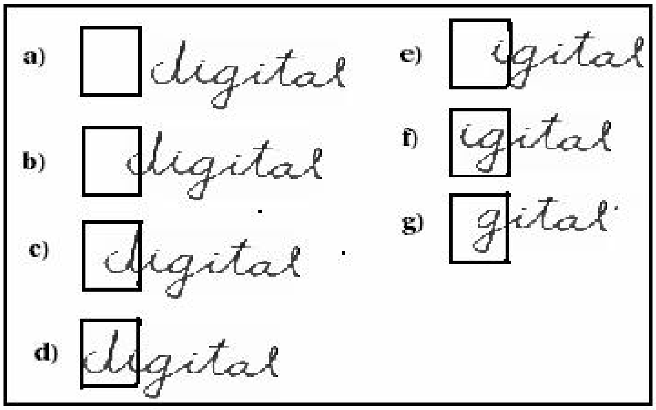

Figure 6: Example of retrospective recognition

The Figures 6(a-g) explains this phenomenon. The word image to be recognized is "digital" (a). In (b) the longitudinal shifter shifts the word to the left once and the recognizer output is is 'c'. On shifting left once more (c), the recognizer output becomes 'd'. In the first two cases, a very high confidence level is achieved. In (d), on shifting once more, the recognizer allots a very low confidence level. Hence the the confidence curve falls down drastically. At this point s definite recognition of a character is said to be made. The recognition system takes the last character with the highest confidence level and decides that it represents the first character image of the word. On definitely deciding a character, the system removes the character image from the word image and the same sequence of steps is followed for the other characters. In (e) the character 'd' has been removed and a high confidence level is allotted to 'i'. At (f) the confidence curve falls down and the second character is definitely decided as 'i' and the character image of 'i' is removed from the word image.

High accuracy in retrospective recognition is augmented by the usage of field specific context sensitive dynamic vocabulary. In the above recognition process, as and when a character is identified, the recognition unit traverses to the corresponding character node in the star search data structure. For instance, after recognizing 'i' in the word "digital", the recognition system would be pointing to the corresponding 'i'th node in the star data structure. At this point, it knows the list of all possible outcomes for the next character. If the recognized character does not result in a valid combination, the recognition unit selects the next highest confidence level match for that character and checks for validity.

Figure 7: a) cleaveland b) deaveland

For instance consider the Figure 7 which shows a place field image. Here the recognition output basically depends on the way the word is segmented. The output can either be "cleaveland" or "deaveland". By using context sensitivity, we can identify that "deaveland" is an incorrect output for a place and we can reject it. Thus by employing context sensitive dynamic vocabulary in the form of a star data structure, very high accuracy levels can be achieved.

6. Conclusion

Once all the imperative fields are recognized, they can be represented in the form of a Delivery Point Code (DPC). [1] proposes a practical way of generating the DPC using Pincode and street address. The DPC can later be converted into a low-cost machine readable format, such as barcode, and can then be printed on to the mail. Once this is done, all further sorting processes can be fully accomplished economically using low-cost barcode readers and mechanical conveyers.

The primary motive behind this work originated from an attempt to achieve total automation of Indian Postal System taking into consideration all the associated bottlenecks. The adaptive line/word separation techniques and the highly robust recognition system used by PMP ensures high degree of accuracy in its processing.

References

[1] P.Sri Rama Prasanna, S.Balaji, Thejavor Haralu Khezhie, C.Vasanthanayaki, S.Annadurai, "*Destination Address Interpretation For Automating the Sorting Process Of Indian Postal System*" in IEEE Tencon – 2003 Region 10

[2] By Jim Terry, Stacy Kamigaki, and Tatyana Vazulina "*Recognizing the Truth about Character Recognition*" Parascript LLC

[3] http://www.languageinindia.com/

[4] Rafael C. Gonzalez, Richard E.Woods, 'Digital Image Processing',Pearson Edication Asia, 1992 edition.