Machine Reading of Tamil Books - An Aid for the Blind

K G Aparna, G L Jayavardhana Rama and A G Ramakrishnan
Biomedical Laboratory, Dept of EE, Indian Institute of Science, Bangalore – 560 012, INDIA
E-mail: {prjocr, prjkss, ramkiag}@ee.iisc.ernet.in

Abstract - **This paper reports on the work carried out so far in an ambitious project of enabling blind people to read any book in an Indian language. It combines optical character recognition of printed text in Tamil and text-to-speech conversion. For a good quality printed document, we have achieved a recognition rate of about 98%. The speech synthesis part of the work is also in an advanced stage, with features such as changes in pitch contour for interrogative and affirmative sentences.**

Index Terms – **blind aid, text-to-speech, optical character recognition, machine reading, waveform concatenation.**

I. Introduction

A machine that can read books in our own languages would be a yeoman service to the blind (visually handicapped) people in empowering them with knowledge. The work reported here deals with the development of such a system that can read books in Tamil. The system can be broadly classified into two major blocks, namely, Optical Character Recognition (OCR) and Text to speech (TTS) conversion. OCR deals with the recognition of printed text and storing it in one of the coding standards like ISCII or Unicode (also TAM or TAB for Tamil). TTS stage involves conversion of the recognised text to speech. Systems such as this are available for English and certain other European languages. However, to the knowledge of the authors, such systems are not available in any of the Indian languages, though one such system is under development on Bangla.

II. Optical Character Recognition

The first step in the whole process is to scan the printed text page and convert it into a digital image. We scan the document at 300 dots per inch. Once the digital document is obtained, the recognition process proceeds as shown in the block diagram in Figure. 1. This has two phases – the training phase and the recognition phase

A. Preprocessing

This involves binarisation, skew detection and skew correction. Binarisation converts the given gray scale image into a binary image. Skew correction is the next important step in OCR. Placing the paper on the scanner may introduce some tilt (skew), or there may be skew in the print itself. We estimate the skew angle in two steps using a precise detection algorithm [1]. A coarse estimate of the skew is obtained through interim line detection using Hough Transform. The interim lines are the lines that bisect the backgrounds in between the text lines. The coarse estimate is used to segment the text lines, which are then superposed on each other and the direction of the principal axis of the resulting image is taken as the fine skew direction. The accuracy of the final estimate is +/- 0.06^0. Skew correction is performed on the original gray level image rather than the binary image to avoid quantisation effects.

B. Segmentation

The next important step is segmentation, which is performed on a skew corrected image. In OCR it is the quality of segmentation that decides whether one can get good or poor classification results. Lines are segmented by smoothing the horizontal projection profile with the help of a Gaussian filter and then finding the minima points. Then individual lines are fed to a run length smoothing algorithm, which is nothing but essentially filling up the gaps within and around the character. By this clusters of words are obtained. Finally taking vertical projection profile on this run length smoothed image results in word segmentation. Connected component analysis is employed to segment characters from words, which are then normalised and thinned to a predefined size. Symbol normalisation is performed in order to bring individual symbol to a normalised size so that they can be compared with those of the known symbols in the reference database. Thinning [2] is performed on this normalised symbol to make the recognition process independent of font and size.

C. Feature Extraction and Recognition

The segmented symbols are sent to the classifier for recognition. It is desirable to divide the set of 154 Tamil symbols into a few smaller clusters to reduce the search space for recognition, resulting in less recognition time and lower probability of confusion. This is accomplished by designing a three level, tree structured classifier to classify the Tamil script symbols.

1) First level classification based on height:
The text lines of any Tamil text will have 3 different segments as shown in Figure. 2. Since the segments occupied by a particular symbol are fixed and generally invariant to font, a symbol can be associated with one of the four different classes depending upon its occupancy of these segments:
Symbols occupying segment 2 alone - Class 0.
Symbols occupying segments 1 and 2 - Class 1.
Symbols occupying segments 2 and 3 - Class 2.
Symbols occupying all the segments - Class 3.

2) *Second level clustering based on Matras.*
This level of classification is applied only to symbols of classes 1 and 2, which have upward and downward extensions (matras). These are further classified into Groups, depending on the type of ascenders and descenders present in the character. This level of classification is feature based i.e. feature vector of the test symbol is compared with the feature vector of the normalised training set. The feature used in this level is second order geometric moments and the classifier employed is nearest neighbour.

3) *Recognition at the third level.*
In the third level, recognition is performed on the normalised symbols, using 2-D discrete cosine transform coefficients as features. A symbol is rejected if the distance to its nearest neighbour in the training set is larger than a predefined threshold. The recognised characters are then stored using TAB codes.

D. Training set / Database

In order to obtain good recognition accuracy, we have created a vast database of size exceeding 4000 samples. Each character has 25 to 50 samples collected from various magazines, novels, and technical papers and from various Tamil shloka books. The database also covers bold and italic characters, as also special symbols like comma, semicolon, colon and numerals. Fonts like TM-TT Valluvar, TAB_Arulmathi, Inaimathi, TM-TT Bharathi and TAM-Aniezhai provided by Tamil editors like Kamban, Murasu Anjal and iLEAP are also included. We have handled font sizes from 14 to 20 in testing the system.

The training set contains the features of the normalised and thinned symbols. The features of the unknown symbol are compared with the sets of known symbol and a label of the one that closely matches is assigned to the test character.

III. Text to Speech Conversion

The techniques employed for synthesizing speech from text may be broadly classified into three categories:

I. Formant-based
II. Parameter-based
III. Concatenation-based

In the formant-based approach [6], we can synthesize a signal by passing the global periodic waveform through a filter with the formant frequencies of the vocal tract. It makes use of the rules for modifying the pitch, formant frequencies and other parameters. However, the technique fails to produce good quality, natural sounding speech, since deriving appropriate rules for unlimited speech is rather cumbersome. As the model uses a number of resonators, it is computationally expensive.
On the other hand, in parameter-based synthesizers [6], the waveforms are modelled using Linear Prediction (LP) coefficients. The linear prediction model is an all-pole model which models vowels exceptionally well, but fails to model the nasals and silence (stops) perfectly.
In concatenation-based speech synthesis, natural speech is concatenated to give the resulting speech output. This is more natural but the database size is fairly huge. Concatenation method can be of three different types:

(a) Limited domain waveform concatenation
For a given limited domain, this approach can generate very high quality speech with only a small number of recorded segments. Such an approach, used in most interactive voice response systems, cannot synthesize arbitrary text. Many concept-to-speech systems use this approach.

(b) Concatenation without waveform modification
Unlike the previous approach, these systems can synthesize speech from arbitrary text. They can achieve good quality on a large set of sentences, but the quality can be mediocre for other sentences where poor concatenations take place.

(c) Concatenation with waveform modification
These systems have more flexibility in selecting the speech segments to be concatenated, because the waveforms can be modified to allow for a better prosody match. This means that the number of sentences with mediocre quality is lower than the case where no prosody modification is allowed. On the other hand, replacing natural with synthetic prosody can affect the overall quality. In addition, the prosody modification process also degrades the overall quality.
In our work, we have synthesized speech using concatenation *with waveform modification* approach, where, naturally spoken speech units are concatenated to give the resulting speech output. This is more natural than the parameter based approach, but the database size is fairly huge. After analysing the performance of the system, we proceed to propose a method to improve the quality of the synthesized speech. Figure 3 gives the block diagram of our complete TTS system using concatenation principles.

A. Offline process

The offline processes of the system include (1) Choosing the basic units (2) Building the database (3) Detailed study of prosody in natural speech (4) Consonant-vowel segmentation and (5) Pitch marking.

B. Deciding the basic units

Basic unit of speech is a phoneme. Other units that can be used for synthesis are diphones, triphones, demi-syllables, syllables, words, phrases and sentences. In terms of the final quality of

speech, sentence is the best unit and phoneme is the worst. However, the size of the database also is an important factor to be considered. Practically infinite units must be stored if the basic units are sentences. The issues in choosing the basic units for synthesis are:

The units should lead to low concatenation distortion. A simple way of minimizing this distortion is to have fewer concatenations and thus use long units such as words, phrases or even sentences. However, as described in the previous paragraph, the size of the database size can become unwieldy.

The units should lead to low prosodic distortion. Whereas it is all right to have units with prosody slightly different from the desired target, replacing a unit having a rising pitch with another having a falling pitch will result in an unnatural sentence.

The units should be of a general nature, if unrestricted text-to-speech is required. For example, if we choose words or phrases as our units, we cannot synthesize arbitrary speech from text, because it is almost guaranteed that the text will contain words not in our inventory. As an example, the use of arbitrarily long units results in low concatenation distortion but it has been shown that over 180,000 such units would be needed to cover 75% of a random corpus. The longer the speech segments are, the more of them we need, to be able to synthesize speech from arbitrary text. This generalization property is not needed if closed-domain synthesis is desired.

Considering the above issues, syllables have been used as basic units. This may contain phonemes, diphones or triphones. The different *instances* of the unit are V, CV, VC, VCV, VCCV and VCCCV, where V stands for a vowel and C stands for a consonant.

C. Building the database

The database was collected from a male, native Tamil speaker over a span of several months. Recording took place in a noise free room. Spoken units were recorded at a sampling rate of 8 KHz.

D. Observation of prosody in natural speech

Prosody is a complex weave of physical, phonetic effects that are employed for expression. Prosody consists of systematic perception and recovery of the speaker's intentions based on pauses, pitch, duration and loudness. Pauses are used to indicate phrases and the ends of sentences or paragraphs [5] . It has been observed that the silence in speech increases as we go from comma to ends of sentences to ends of paragraphs. Also duration is an important factor that affects the naturalness of the synthesized speech. The same vowel has different durations when it occurs in different positions in words or sentences. For example, consider the sentence "/naan aaru manikku varalaamaa?/". In this sentence, vowel /aa/ has different durations at different positions as tabulated in Table 1.

Duration analysis is performed on a set of samples recorded from a native Tamil speaker. The information is tabulated and stored as a look-up table for future reference. Although loudness is not as important a phenomenon as pitch, it may introduce artifacts. The artifact is in the form of an echo. This is caused due to the amplitude envelope mismatch.

E. Consonant Vowel Segmentation

It is observed that any change in the consonant part of a signal results in a change of perception of the unit. Consonants must be kept intact. To this end, consonant and the vowel regions of the units must be segmented. In terms of morphological structure, consonant can be classified into co-articulated and non co-articulated signals.

Non co-articulated consonants can be segmented easily using the difference of energy between consecutive blocks of the signal. The given speech unit is divided into frames of 10 msec duration each. Energy of each frame is calculated and the first difference of the energy contour gives two distinct peaks, one on the positive side and the other, on the negative side.

Segmentation of co-articulated consonant is more challenging. The energy contour is almost flat at the transition from vowel to consonant. Preliminary results based on spectral analysis are available. For completion of the system, we resorted to manual segmentation.

F. Pitch Marking

Pitch marking is essential as the waveforms are concatenated at the pitch marks. Unbiased autocorrelation [7] is employed to get distinct peaks at the pitch frequency. After getting the peaks, nearest zero crossing to the left of the peak gives the pitch mark. The results of pitch marking and segmentation of non-co-articulated and co-articulated consonants are shown in Figures. 4 and 5, respectively.

G. Online Process

This has two phases: Text analysis and Synthesis. Text analysis comprises parsing the input text into a sequence of basic units of speech and application of Tamil rules. Synthesis consists of concatenation of the waveforms of these units in the correct sequence and application of the prosodic rules. Prosodic rules modify the duration and amplitude. Correlation is employed to minimise concatenation artefacts. A small variation in pitch at the concatenation point does not affect the quality of speech. From the pitch marks obtained by amplitude matching, five pitch periods are correlated to get the phase matching point. Amplitude mismatches at points of concatenation cause echo-like sounds. This is normalised by fixing a threshold and matching the amplitudes of the vowels.

IV. IMPLEMENTATION

The system is designed to work on Windows 95 and Windows 98. It is designed using C++ and graphic user interface (GUI) is provided using Visual C++. OCR approximately takes two minutes for a scanned A4 page containing around 1200 characters on a 500 MHz Pentium III machine with 128 MB RAM. Speech synthesis takes a further 15 seconds.

V. CONCLUSIONS

The product has been tested on different fonts and the recognition rate is around 98% with the presence of some special characters and numerals. When this text is input to the TTS, most of the errors in recognition are masked due to the knowledge of the listener. Attempts are being made to improve the accuracy in recognition. Work is underway to reduce the database for speech synthesis system and also synthesis of emotions is being attempted.

VI. ACKNOWLEDGEMENT

The authors thank Ministry of Information Technology, Govt. of India and Tamil Software Development Fund, Govt. of Tamilnadu for funding parts of the work reported here.

VII. REFERENCES

[1] Kaushik Mahata and A. G. Ramakrishnan (2000). A complete OCR for printed Tamil text, *Proc. Tamil Internet 2000, Singapore (July 22-24, 2000)*, 165-170.

[2] R C Gonzalez and R E Woods (1999). *Digital Image Processing*, Addison – Wesley Press, New York.

[3] G Strang, *Linear Algebra and its Applications*, Academic press.

[4] R O Duda and P E Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons.

[5] Xuedong Huang, Alejandro Acero, and Hsiao-Wuen Hon. *Spoken Language Processing*, Microsoft Press.

[6] Douglas O'Shaughnessy (2000), *Speech Communication*, IEEE Press, New York.

[7] Rabiner L R (1977). On the use of autocorrelation analysis for pitch detection, *IEEE Trans. Acoustics, Speech, Signal Processing ASSP-25*, 24-33.

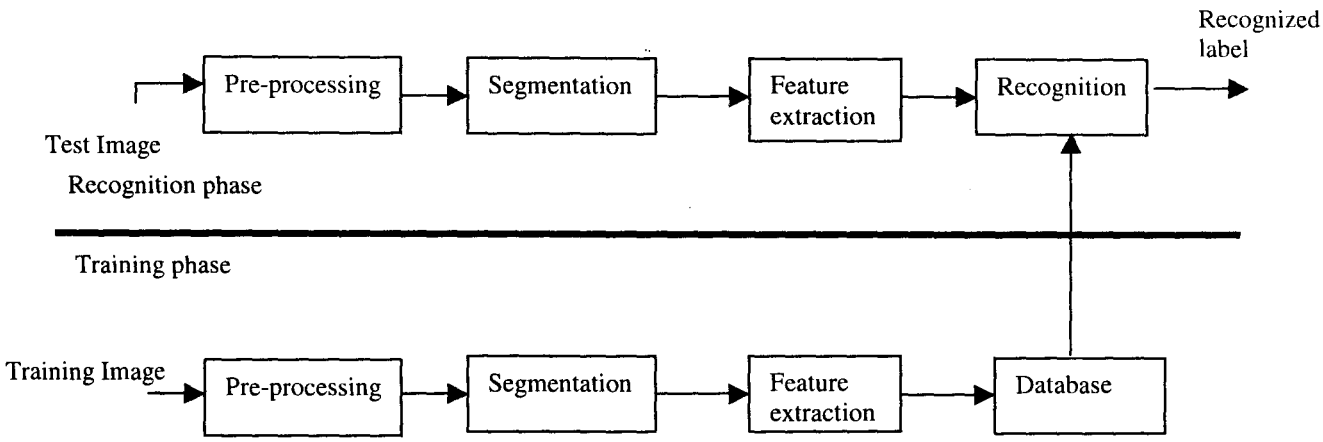

Figure 1. Block Diagram of the OCR System.

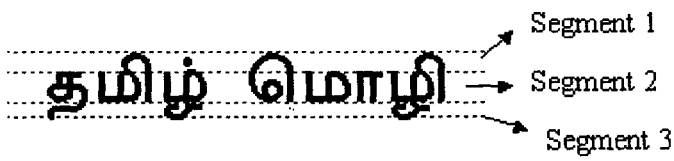

Figure 2. Result of four-line detection

	Position	Duration (Sec)
Naan	Middle	0.15
Aaru	Initial	0.16
Varalaamaa	Middle	0.15
Varalaamaa	Final	0.18

Table 1:Duration of vowel /aa/ when it appears at different positions

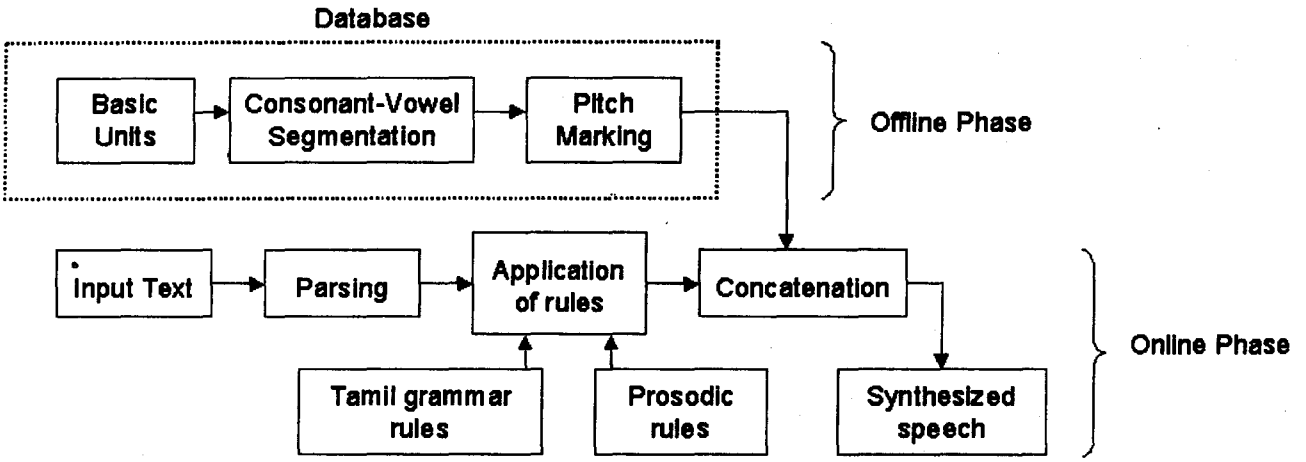

Figure 3. Block diagram of speech synthesis system using concatenation

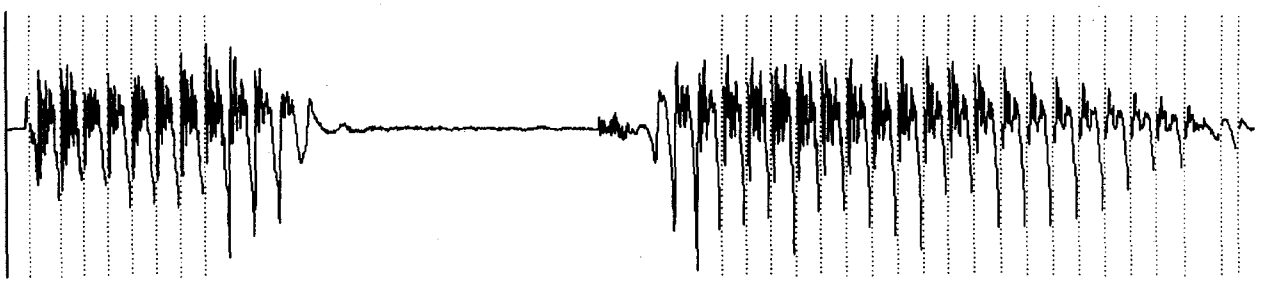

Figure 4. Segmented and pitch marked non-co-articulated VCV /aka/

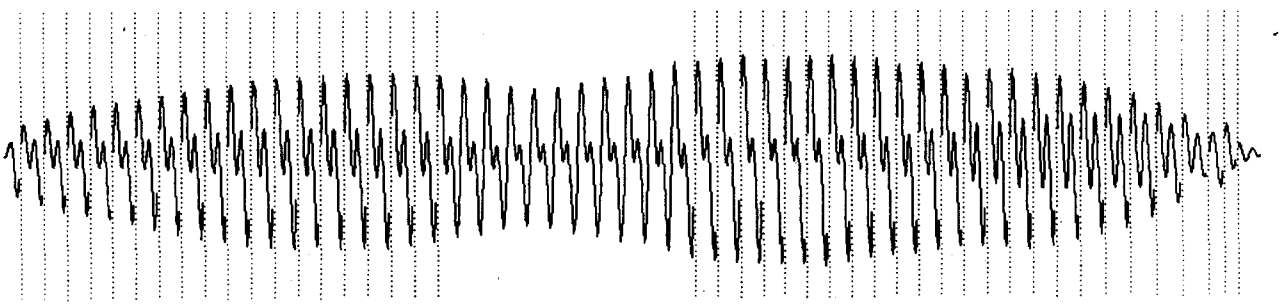

Figure 5. Segmented and pitch marked co-articulated VCV /iyi/