9

PAARVAI – Book Reader for the Blind

A G Ramakrishnan, Ashwini C M, Jayavardhana Rama and G Sita

Biomedical Engineering Laboratory, Department of Electrical Engineering,
Indian Institute of Science, Bangalore – 560 012.
<E-mail: sita@ee.iisc.ernet.in, ramkiag@ee.iisc.ernet.in>

Abstract

The ambitious project of facilitating blind people to read a Tamil book by themselves, without others' help has been addressed. This paper addresses on the work carried out till date. It is the combination of optical character recognition of printed text in Tamil and text-to-speech conversion. It is a preliminary version, which can read small paragraphs of printed text in Tamil. The demo version, running on Windows platform will be demonstrated in the conference.

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

Introduction

The global revolution in the field of technology and communication has made an impact in the life of the differently abled. In the near future, a blind man will be able to read a book with the help of a machine. Our project is a major step in this direction. The work reported here deals with the development of a comprehensive system that can read a Tamil printed book.

The system consists of 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 TAM or TAB for Tamil. TTS does the work of converting the recognized text to speech.

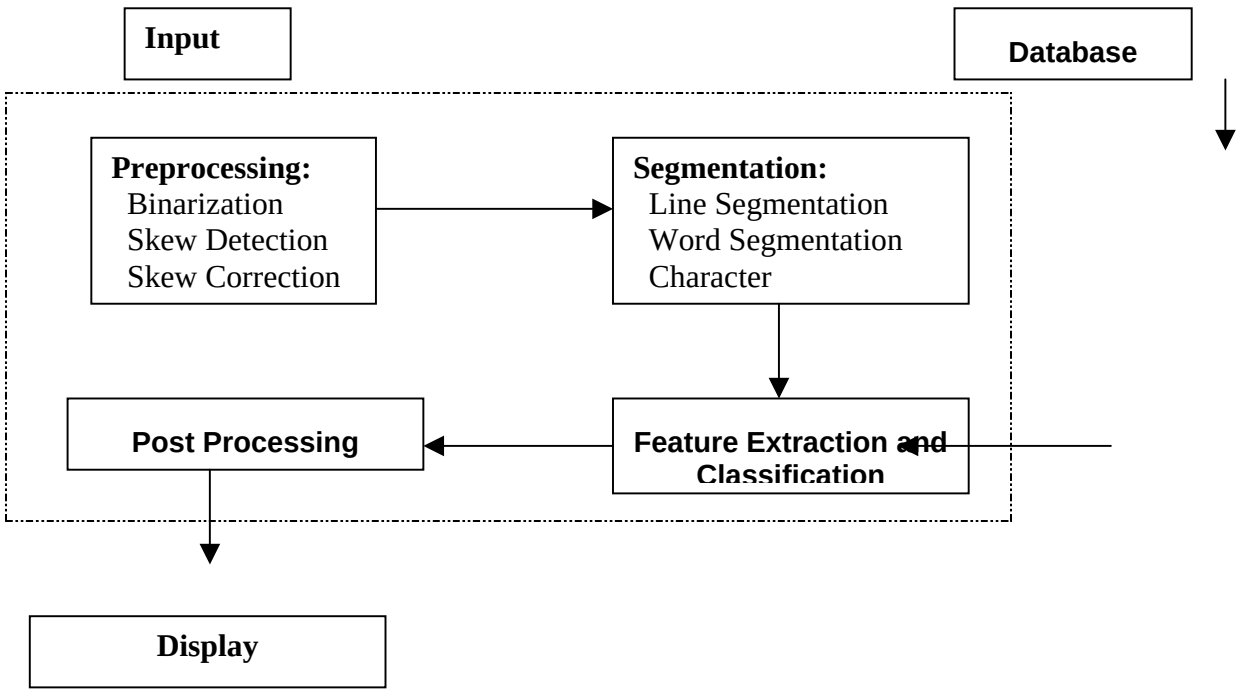
Optical Character Recognition

The initial step in optical character recognition is to scan the printed text page and digitise it. The scanning of the document is performed at 300 dots per inch. Once the digital document is obtained, the recognition process proceeds as shown in the block diagram.

Preprocessing

The preprocessing is performed through binarisation, skew detection and skew correction. Binarisation converts the given gray scale image into a binary image. Placing the paper on the scanner inappropriately introduces tilt (skew), or there might be skew in the print by itself. The skew angle is acquired by skew detection. 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. The skew correction removes this undesirable skew. It is performed on the original gray level image rather than the binary image to avoid quantisation effects. The skew detection and correction forms the critical step in OCR.



Block Diagram of Optical Character Recognition

2. Segmentation:

The next important step is segmentation. The evaluation of OCR depends on the result of segmentation. The segmentation is done by line segmentation, word segmentation and character segmentation. The line segmentation is performed by smoothing the horizontal projection profile by a Gaussian filter and detection of the minima points. The individual lines obtained are fed to a run length-smoothing algorithm, which fill the gaps within and around the character. The result is a cluster of words. Subsequently, taking the 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 is performed on this normalised symbol to make the recognition process independent of font and size.

Feature Extraction and Recognition

The segmented symbols are sent to the classifier for recognition. The Tamil dataset is made up of 154 symbols. It is preferable to divide this set into a few smaller clusters to reduce the search space for recognition. This results in less recognition time and lower probability of confusion.

The division is accomplished by designing a three level, tree structured classifier to classify the Tamil script symbols:

First level classification based on height:

The text lines of any Tamil text will have 3 different segments. 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 as depicted below:

Segments Occupied	Symbol Class
Segment 2 Alone	0
Segment 1 and 2	1
Segment 2 and 3	2
All Segments	3

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.

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.

Post Processing

In this stage we try to distinguish characters that generally get confused. Based on certain set of language rules, we try to correct the misclassified character. The table below shows a few confusing characters.

ழூ	ழூ
ஜ	ஜ
ழு	ழு
ஶ	ஶ
ந	ற
பூ	யூ
யு	பு

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.

Text to Speech Conversion

The second major block of this project is the Text-to-Speech synthesis. The technique employed for synthesizing speech is based on concatenation with waveform modification. In this technique, natural speech is concatenated to give the resultant speech output. This is more natural but the database size is fairly huge. The Figure below gives the block diagram for the complete TTS system.

The TTS is made up of the Offline process and the Online process.

Offline process: The offline process includes:

1. Text Corpus:

The text corpus consists of phonemes, which are the basic units of speech. Other units that can be used for synthesis are diphones, triphones, demi-syllables, syllables, words, phrases and sentences. The condition for choosing the basic units is that *the units should have low concatenation distortion* and *the size of the database.* In terms of the final quality of speech, sentence is the best unit and phoneme is the worst. However, infinite units have to be stored if the basic units are sentences. The other issues for the selection of the basic unit are *the units should lead to low prosodic distortion and should be of general nature.* Considering all the above conditions, syllables have been used as the basic units in our project. 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.

2. 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.

3. 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. It has been observed that the silence in speech increases as we go from comma to ends of sentences to ends of paragraphs. The duration too 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.

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.

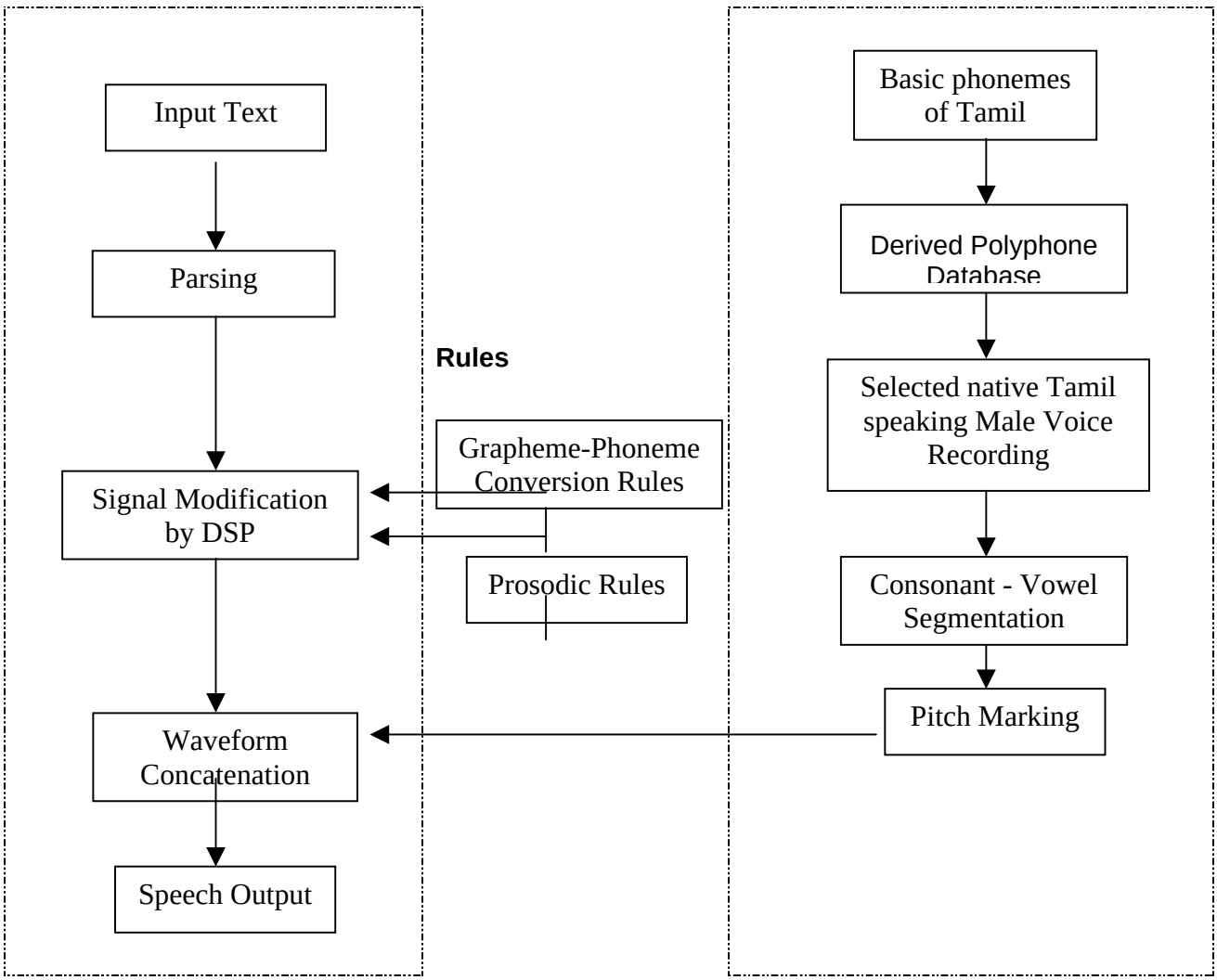
Online Phase **Offline Phase**

Block Diagram of Text-to-Speech Synthesis

4. 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. Therefore, care should be taken to keep the consonants intact. To this end, consonant and the vowel regions of the units must be segmented.

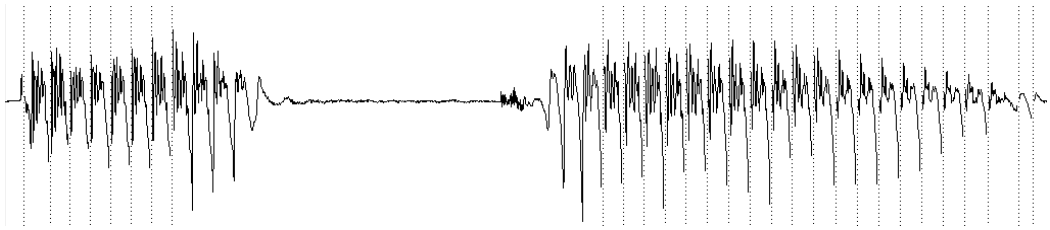
In terms of morphological structure, consonant can be classified into co-articulated and non co-articulated signals. Non co-articulated consonants can be segmented easily using the difference of energy between consecutive blocks of the signal. The given speech unit is divided into



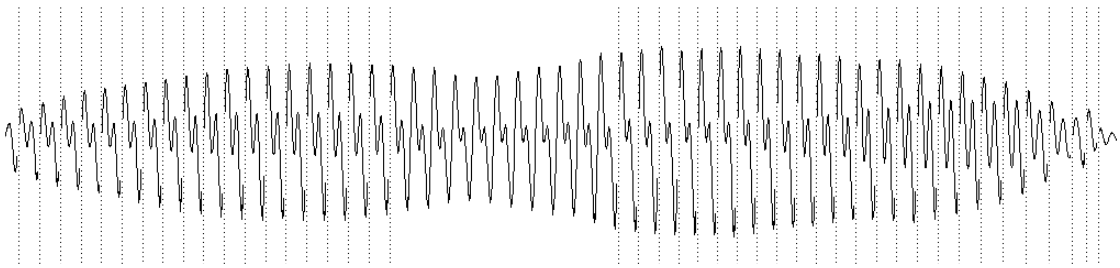
frames of 10 ms duration each. Energy of each frame is calculated and the first difference of the energy contour gives two distinct peaks, one on the positive side and the other, on the negative side. Segmentation of co-articulated consonant is more challenging. The energy contour is almost flat at the transition from vowel to consonant. Preliminary results are based on spectral analysis. Further, we resorted to manual segmentation.

Pitch Marking

Pitch marking is essential as the waveforms are concatenated at the pitch marks. Unbiased autocorrelation 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 below respectively.



Segmented and pitch marked non-co-articulated VCV /aka/



Segmented and pitch marked co-articulated VCV /iyi/

Online Process

The online process has two phases:

Text analysis

Text analysis comprises parsing the input text into a sequence of the basic units of speech and application of Tamil rules.

Synthesis

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 minimize concatenation artifacts. 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 normalized by fixing a threshold and matching the amplitudes of the vowels.

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.

Conclusions

The product has been tested on different fonts and the recognition rate is around 95% with the presence of some special characters and numerals. When this text is input to the TTS, some 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.

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.

10

Internationalization of the Domain Name System: The Next Big Step in a Multilingual Internet

Tan Tin Wee and S. Maniam

National University of Singapore, Singapore 119260
i-DNS.net International Inc, Menlo Park, California, USA
Email: s_maniam@pacific.net.sg

Abstract

Every machine on the Internet has an Internet address (IP address). The Internet domain name provides a meaningful and easy-to-remember handle for an IP address. The task of mapping domain names to IP address for all the Internet address is performed by the worlds, most extensive and scaleable database system, the Domain Name System (DNS). Almost every common Internet application calls on the DNS to resolve a domain name into an IP address.

Limitations in the DNS and its operation by convention restrict the characters used in domain names to A-Z, a-z, 0-9 and - of the ASCII Latin set. Multilingual characters are not supported. Therefore, even though the content of a webpage or an email may be in a native script of a non-English language, the address cannot be rendered in that native script.

The process of supporting multilingual script and other linguistic and cultural needs on the Internet is generally known as Internationalization. The Internationalization of the DNS system is potentially a gargantuan task of upgrading the DNS protocols that use domain names, which is practically every commonly encountered Internet application, and an overhaul of the DNS servers currently installed worldwide. This process, first initiated in 1998 by a Chairman's Commission of the Asia Pacific Networking Group (APNG), led to a standards working group in the Internet Engineering Taskforce (IETF) and an international consortium comprising industry, academia and regulatory authorities called the multilingual Internet Names Consortium (MINC). Meanwhile, more than a dozen alternative multilingual DNS technologies, known generally as iDNS, have emerged and starting in late1999, the first company, i-DNS.net International Inc., deployed in Chinese and by early 2000, Tamil was the second language to be commercially launched. Since then multi-lingual domain names, now known as IDNs or Internationalised Domain Names, have been deployed in over 60 languages with Asian, Middle Eastern, Western and Eastern Europe, Latin American origins. The advent of the mass deployment IDNs by Verisign - the company that controls the 30 million ASCII "dot com" and 'dot net' domain names amounting to more than half of all domain names in use today - in late 2000 in over 60 languages, the number of IDNs registered has exceeded 1 Million. Today, between Verisign and a number of other entities that have deployed IDNs, this number is approaching 2 million, with the majority coming from the Chinese, Japanese and Korean languages. With the recent finalisation of the IDN standard by IETF in June 2003 and the announcement since by MINC of

an Interoperability Testbed to ensure compatibility of the different deployments centered around the released IDN standard, not only is the number of IDNs in use expected to grow dramatically in the coming year but the related nascent deployments of multi-lingual email addresses is also set to explode by mid-2004.

1. Multilingual Limitations of the current DNS system

The Internet Domain Name System (DNS) was developed over time to facilitate easy recall of Internet address. It achieved this via the matching of easily remembered alphanumeric strings such as www.yahoo.com to the string of numbers of Internet Protocol (IP) address (version 4) eg. 137.132.19.1. Unfortunately, in the pursuit of universality, only ASCII alphanumeric characters plus the- were acceptable characters in the domain name strings. Subsequently, because of the replacement of meaningless numbers with meaningful memorable words, this DNS convention was universally adopted as the global standard for all hostnames, email address, Web addresses, e-mail address and other Internet addressing formats. For instance, until now, no provision has ever been made to allow input domain names in Web address in a non-ASCII non - English script. This meant that any user of the Internet had to have some basic knowledge of ASCII characters (i.e. English language and languages which use basic Latin characters).

While this does not pose a problem to the scientific, technical or business user who is able to understand English as an international language of science, technology, business and politics, it is a major stumbling block especially in countries where English is not widely spoken. This hindrance contributes significantly to the digital divide as the Internet is becomes the standard way for doing business, and slows the popular adoption of the Internet in the business community, particularly in countries struggling to keep up with the pace of development in developed countries.

What has meant to be a memorable way of reaching a website to the English-speaker, has now turned out to be a linguistic barrier to the Internet. In many instances, it may even be easier for a non English-speaking person, say, in village in China, to remember a string of numbers rather than a series of unfamiliar and unintelligible Roman alphabets.

In countries such as these, understanding English has become a daunting prerequisite to performing such as basic activities as posting e-mail or accessing Web pages that are otherwise composed in the local language. The irony is that most software applications today already support a robust local language environment, but access to the Internet, whether email or websites, invariably requires use of ASCII alphabet.

The Internet already encompasses a network of communities representing a global mosaic of languages and cultures. The increasing volume of modern business, research, and interpersonal communications in non-English languages is a testament to this fact. Clearly, the existing DNS has become an anachronism in an already multilingual Internet world.

2. Internationalization of the Domain Names System

In response to the multilingual demands created by the natural evolution of the Internet, the process of internationalization of the Domain Name System (iDNS) had begun.

The task to upgrade the DNS is very complex. The DNS protocol impacts on many Internet protocols. Common protocols such as those supporting the web and email all involve the DNS. Client applications such as Web browsers and email clients have to be changed. The server software in DNS servers throughout the world has to be modified as the DNS system is arguably the world's most extensive, hierarchal, distributed and scaleable database.