Proc. National Conference on Biomedical Engineering 1998 (NCBME 1998), Manipal, Karnataka, India.

A TEXT-TO-SPEECH CONVERTER FOR READING AID FOR THE BLIND

Sachin S. Mayekar, P.C. Pandey, & R.K. Shevgaonkar

Department of Electrical Engineering
Indian Institute of Technology, Bombay
Powai, Mumbai 400 076

Abstract

A reading aid for the blind consists of a scanner for obtaining the digital image of the text, a system for segmenting and recognizing the characters of the text, a converter from text character stream to phonemic representation stream, and a speech synthesizer. A character segmenter-recognizer and a speech synthesizer, for printed Hindi text in Devanagari, has been developed.

Text image processing identifies the characters and "*matras*" in the image file obtained from a scanner. The image is segmented into basic characters and *matras*. Segmentation is done on the basis of pixel density in the hierarchy of lines, word boundaries, characters, and letters and *matras*. These segments are identified by a character recognition algorithm, which compares the segmented image with a set of templates, and we get a stream of character codes, which is converted into a stream of allophone codes. Speech synthesis is achieved by using a synthesizer chip SPO-256-AL2, that internally uses a formant based synthesis and has formant tracks for 64 English allophones. The control of the synthesizer chip is handled by an inexpensive microcontroller (AT59C2051), with a serial port interface for connecting to a computer. A program helps in selecting the appropriate set of English allophones to correspond to the allophones of Hindi.

Introduction

A text-to-speech conversion system is useful in interaction with computers and as a reading aid for the blind. This system typically consist of a scanner for obtaining a digitized image of the text, a system for segmenting and recognizing the characters of the text and a speech synthesizer for generating speech output.

The complexity of the character segmentation-recognition system and speech synthesis depends to a large extent on the complexity of the script system, script symbol-to-phonemic correspondence, and the linguistic rules. In the Roman character set used for writing English, segmentation of characters in the printed text is relatively easy as the characters are separated horizontally by a certain minimum spacing. However, converting the script characters to the combination of corresponding sounds is an involved task. As a result of years of research and development activity, there are a number of character scanners and speech synthesizer systems available for English. These can be put together to form a text-to-speech converter.

Unlike in English, in most Indian languages, the script system is relatively complex, each character being a combination of a basic letter and a "*matra*" corresponding to vowel. As most Indian languages use nearly a phonemic transcription system, the rules for converting the characters into corresponding sounds is less complex as compared to English. However, special care has to be taken in speech synthesis to get natural sounding speech. We have developed a Devanagari script based text-to-speech conversion system for use with Hindi and it can be extended to other languages like Marathi, Nepali, Sanskrit.

The system, as shown schematically in *Fig.1*, consists of a computer, an image scanner and a speech synthesizer. The image scanner, connected to the computer gives an image file of the document with text in Devanagari. The script is a character sequence, which is converted into speech output using a speech synthesizer, as per the linguistic rules of Hindi. We have opted for "allophone" synthesis, as it poses no restriction on the set of words or sentences that can be handled. In our system, a commercially available scanner is used. The software for

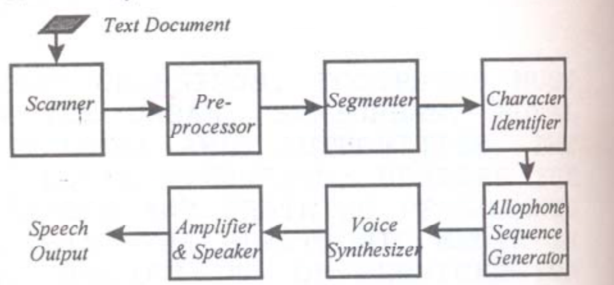


Fig. 1 Processing Sequence in Text-to-Speech Conversion

character segmentation and identification, and the hardware for speech synthesis has been developed.

Text Image Processing

Text image processing involves identifying the constituents of the text image obtained by a scanner at a resolution of 300 to 600 dpi. The image file is first pre-processed to remove noise.

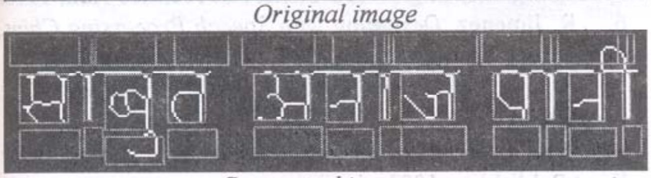
Pre-processing

The raw image is pre-processed by filtering (cleaning), thresholding and thinning. Weighted mean filtering [1] removes unwanted noise points present in the scanned image. Thresholding binarizes the grey level image obtained after cleaning. Thinning retains only the skeleton of the cleaned and binarized text image, as thickness of the lines does not convey information about a letter. It is implemented by "median axis transform" method [1,2].

Segmentation

The text image contains a character stream to be identified. The image segmentation is done in following sequence: sentences, words, characters, letters and *matras* as shown in the example in *Fig. 2*. Letters and *matras* are the basic constituents of Hindi. Segmentation creates separate images for these basic constituents from the whole text image. The segmentation is based on pixel histogram of the binary image, in vertical and horizontal directions [3]. Histogram of the image in the vertical direction is a projection of pixels along all horizontal pixel streams on a vertical line. This vertical projection segments the image into lines. The pixel sum is maximum at the middle of a line and reduces towards the *matras* on both sides of the line. This histogram shows three distinct regions. These regions correspond to the lines, space between sentences and *matras* on both sides in a sentence. An appropriate threshold applied on the histogram isolates the line and the *matra*. Each line is then segmented into words, as the pixel density in between the words is almost zero. Words are separated into characters by applying histogram in the horizontal direction.

Original image



Segmented image

Fig. 2 *Example of segmentation of pre-processed image*

Character Identification

Character identification is done in two steps: recognition of letters and association of *matras* with the letters. For letter recognition, segmented pattern of the letter is compared with a set of templates (skeletons of the standard letters and *matras*) by using an artificial neural network based program [4], which gives a matching score. The template giving the best score is taken as the identified character. Association of *matras* with letters is done on the basis of previous and next pattern present on either side of the letter. Unlike the template matching for letters, recognition of *matra* is done on the basis of variation of black and white pixels in the *matra*, which are typical for each type of *matra*. In Hindi, *matra* associates an appropriate vowel to the identified letter.

Generation of Allophone Codes

Identification of letters and *matras* result into a stream of character codes. An appropriate conversion of character codes into an allophone stream needs an application of linguistic rules of Hindi. A program has been developed for mapping the characters into allophones without considering the typical stresses on allophones. This works well for simple words. However, for natural sounding speech appropriate stresses have to be provided.

Programs have been developed for pre-processing, segmentation and character identification. The codes were written with special consideration for memory management for handling large image files on DOS.

Speech Synthesizer

For voice output of a limited set of messages, it is possible to have digitized speech signal compressed and stored for various words and phrases. By combining these, it is possible to generate speech. Such a voice output system is very restrictive for use in a text-to-speech converter. Speech is formed by the combination of basic units called "phonemes". Depending on the context, the same phoneme may have different articulatory parameters associated with it. These different forms of the same phoneme are known as allophones. The phonemes and allophones are language dependent. In articulatory and formant synthesizer, speech is generated from the tracks of a set of articulatory or formant parameters. This gives a great deal of flexibility in selecting the voice characteristics [5]. However the total number of parameters involved is very large, making the task of parameter generation very difficult. In an allophone synthesis, parameters corresponding to different allophones are stored and allophones are selected for synthesis based on the text representation. This results in a great deal of reduction in data to be stored and transferred [6].

We have developed a hardware based allophone speech synthesizer, which can be easily controlled by a computer using an interface which is independent of internal bus architectures and operating system. Based on the cost and availability we have used the syntheiszer SPO-256-AL2 [6]. This IC contains a set of 64 allophones, internally stored on its ROM, for speech synthesis in English. It is an allophone speech synthesizer which uses formant coding for synthesizing its allophones. We have experimented with its allophone set for adopting it for use in Hindi.

The speech synthesizer chip requires allophone addresses as eight parallel lines. An interface based on asynchronous serial port has been developed, so that interfacing remains independent of the bus architecture of the computer. The data transfer rate of serial port was adequate for this purpose. The controller for the synthesizer should take care of all timing consideration in address loading. Microcontroller Atmel AT59C2051 [7] (instruction set compatible to Intel MCS-51 [8]) has been used for the purpose. The hardware is schematically shown in *Figure 3.* The address lines along with address load (ALD) are used for selecting the allophone to be synthesized. On application of ALD, the synthesizer generates the speech and outputs it as a pulse width modulated waveform, which is low pass filtered to get the analog speech output. Sending appropriate sequence of allophone codes result into generation of words and sentences. The timing is achieved by checking the load request (LRQ) line which activates after synthesis of every allophone. The program on the microcontroller, arranges its memory as a circular buffer, which stores addresses received from the computer. To control the synthesizer from the PC port a graphical user interface has been developed which helps in selecting appropriate allophones from the synthesizer internal ROM.

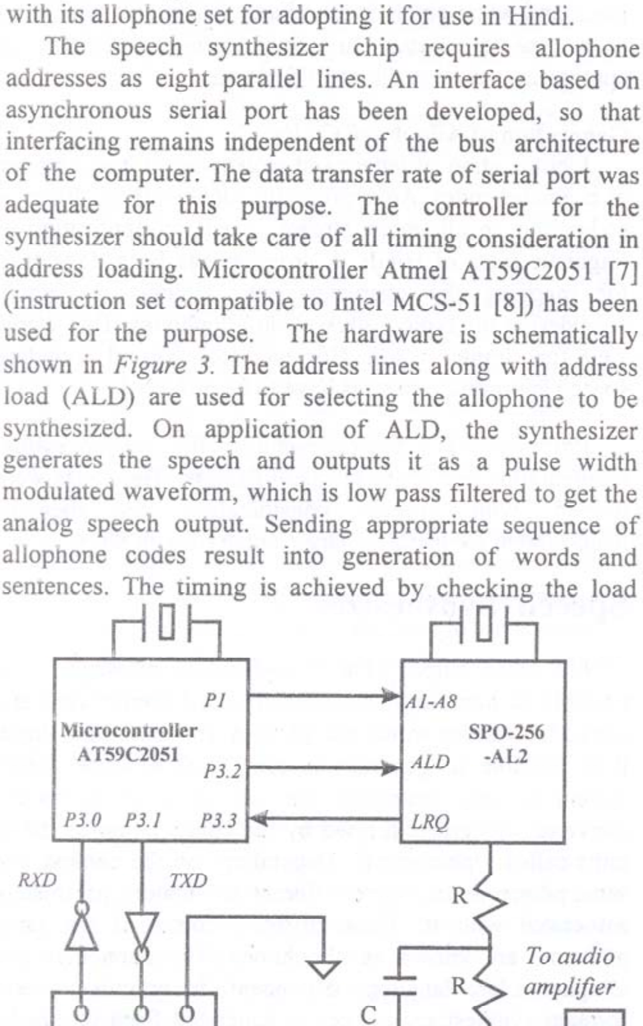


Fig 3 Hardware schematic of the speech synthesizer

Results and Discussion

It takes about 4-5 minutes on a PC with a Pentium 100 MHz processor to process one A4 page image containing 10 lines with 30 characters per line. Text image processing system works satisfactorily for printed Hindi text with a uniform font size in individual lines and no *yuktakshar* (joint letters). The synthesizer chip used has on-the chip 64 English phonemes, and appropriate ones have been selected for approximating Hindi allophones. Some allophones cannot be approximated due to unavailability of similar allophones in English. Also quality of some of the allophones like /r/ is not natural sounding. To improve the quality of synthesized speech, parameters tracks for synthesis should be worked out and provided as a ROM, and the synthesizer should be re-designed to use these tracks.

All the software programs and the hardware can be used independent of each other, or they can be integrated to form a Hindi text-to-speech conversion system.

References

1. A.K. Jain, *Fundamentals of Digital Image Processing,* Englewood Cliffs, NJ: Prentice Hall,1993.
2. D. Philips, *Image Processing in C,* New Delhi: BPB, 1995.
3. U. Pal & B. Chaudhari, "Computer Recognition of Printed Bangla Script," *Int. J. of System Science,* Vol. 26: pp 2107-2123, 1995.
4. Kshitij Sharma, *Text Recognition of Devanagari Script,* B.Tech. Dissertation, Guide: R.K. Shevagaonkar, Electrical Engineering Dept., IIT Bombay, 1997.
5. L.R. Rabiner & R.W. Schafer, *Digital Processing of Speech Signals,* Englewood Cliffs, NJ: Prentice Hall, 1978.
6. R. Jimenez, *Designing with Speech Processing Chips,* San Diego, Cal: Academic, 1991.
7. *Atmel Microcontroller Data Book,* San Jose, Cal: Atmel Corp. 1995.
8. K.J. Ayala, *The 8051 Microcontroller,* St. Paul, MN: West Publishing, 1991.