Text to Speech Conversion with Phonematic Concatenation

Tapas Kumar Patra  
Reader, Dept of Instrumentation & Electronics Engineering  
College of Engg. & Technology, Bhubaneswar, India

Biilab Patra  
Dept of Instrumentation & Electronics Engineering  
College of Engg. & Technology, Bhubaneswar, India

Puspanjali Mohapatra  
Assistant Professor, Dept. of Computer Science and Engineering  
IIIT, Bhubaneswar  
Bhubaneswar, India

Abstract—This paper presents a method to design a Text to Speech conversion module by the use of Matlab by simple matrix operations. Firstly by the use of microphone some similar sounding words are recorded using a record program in the Matlab window and recorded sounds are saved in .wav format in the directory. The recorded sounds are then sampled and the sampled values are taken and separated into their constituent phonetics. The separated syllables are then concatenated to reconstruct the desired words. By the use of various Matlab commands i.e. wavread, subplot etc. the waves are sampled and extracted to get the desired result. This method is simple to implement and involves much lesser use of memory spaces.

Keywords-Text to Speech Conversion; Phonematic Concatenation; Sample.

I. INTRODUCTION

Text-to-speech (TTS) convention transforms linguistic information stored as data or text into speech. It is widely used in audio reading devices for blind people now a days [1].In the last few years however, the use of text-to-speech conversion technology has grown far beyond the disabled community to become a major adjunct to the rapidly growing use of digital voice storage for voice mail and voice response systems. Also developments in Speech synthesis technology for various languages have already taken place [2][3].

Many speech synthesizers using complex neural networks have also been designed [4]. In the bigger picture, the module can open up a window of opportunities for the less privileged paving the way for a plethora of employment opportunities for them in the job sector. It can also play a defining role in establishing communication of the blind if it is incorporated into mobile phones (so that text messages could be converted into speech). [5][6].

Potential applications of High Quality TTS Systems are indeed numerous. Here are some examples:

A. Telecommunications services

TTS systems make it possible to access textual information over the telephone. Knowing that about 70 % of the telephone calls actually require very little interactivity, such a prospect is worth being considered.

B. Language education

High Quality TTS synthesis can be coupled with a Computer Aided Learning system, and provide a helpful tool to learn a new language. To our knowledge, this has not been done yet, given the relatively poor quality available with commercial systems, as opposed to the critical requirements of such tasks.

C. Aid to handicapped persons

TTS Systems can really act as an aid for handicapped persons particularly the blind persons and can give them a support to actually compete, excel and actively participate in the main stream society.

D. Talking books and toys

The toy market has already been touched by speech synthesis. Many speaking toys have appeared. High Quality synthesis at affordable prices might well change this.

E. Vocal Monitoring

In many cases, oral information is more efficient than written messages. The idea of speech synthesizers in measurement and control systems is getting stronger as oral information is sometimes more efficient than visual information.

F. Multimedia, man-machine communication

In the long run, the development of high quality TTS systems is a necessary step (as is the enhancement of speech recognizers) towards more complete means of communication between men and computers. Multimedia is a first but promising move in this direction

II. VARIOUS SPEECH TERMS

*Phoneme* - A phoneme is an abstract unit that represents sounds and writing in a systematic, unambiguous way [7]

*Phonetics* - It is the representation of the sounds of a language.

*Grapheme* - It is the smallest unit of written language. Multiple graphemes may represent a single phoneme.
III. FEATURES
The main features of the technique applied here for text to speech conversion are as follows:-

1. GUI implementation of the Matlab Module[8].
2. Recording of voice samples through microphone.
3. Phoneme extraction by the use of spectrogram
4. Concatenation of phonemes to create any desired word.
5. Comparison of concatenated word with the original word.

IV. METHODOLOGY

When a sound is recorded, its spectrogram i.e. time domain representation can be viewed. By studying the various amplitude (power) peaks in the spectrogram, the region of presence of phonemes can be detected.

V. RECORDING SOUND USING MATLAB CODE

This program is intended to simplify the recording and basic editing of speech waveforms as well as to present the spectrograms and the time waveform in a side-by-side format for ease of analysis. Microphone is connected initially to the microphone input on the sound card. Next a sound is recorded via a microphone. Sounds recorded get stored in Matlab directory with .wav extension. Sound recording is initiated through the Matlab graphical user interface (GUI) by clicking on the record button. The duration of the recording can be adjusted to be anywhere from 1 to 6 seconds. Once recorded, the time data is normalized to maximum amplitude of 0.99 and displayed on the upper plot in the GUI window. In addition to the time domain waveform, a spectrogram is computed using Matlab’s built in spectrogram function (part of the signal processing tool box).

VI. BRINGING THE SOUNDS INTO MATLAB WORK DIRECTORY

Using the ‘wavread’ commands sounds of .wav extension stored in the Matlab directory, can be imported into Matlab workspace. The sounds imported into the workspace are a sampled version. Many other details like sampling frequency, number of bits per sample, etc. can also be stored.

Sample Code:

[y,Fs1,Ns1]=wavread('pain.wav');
Fs1;
Ns1;

This code stores the samples of the imported .wav file in the variable y; sampling frequency and number of bits in each sample are stored in Fs1 and Ns1, respectively and is displayed by this code.

VII. EXTRACTION OF PHONEMES

There are 44 phonemes in English language were taken into account. The selection of 44 phonemes were based on the list from the source namely Orchestrating Success in Reading by Dawn Reithaug (2002) under National Right to Read Foundation. The task was to extract out the sounds pertaining to these 44 phonemes which forms as a base for creating any English word in a standard lexicon.

The samples of sounds imported into the Matlab workspace are stored in a column vectors. All samples of the imported sound contribute to a particular sound intensity depending on its amplitude, hence, power. The combination of all these components yields the total sound (of the phoneme). So, a few consecutive samples could be selected and then stored in another vector and played using the ‘sound’ command. Therefore, by hit and trial method various desired sounds (phonemes) could be extracted from the imported .wav files.

Sample Code:

[y,Fs1,Ns1]=wavread('pain.wav');
yAIN=y(12000:1:16000);
Traditionally there are three general approaches to convert a phonetic transcription into an acoustic signal. They are articulatory synthesis, Synthesis by rule and synthesis by concatenation [9]. The approach used here is a concatenative one. Most high quality speech synthesizers today are concatenative synthesizers. In a concatenative system, a person records speech containing a large set of basic sound units, usually corresponding to a relatively short sequence of phonemes. The units are excised from the speech, and in most systems, the units are processed with some type of speech coding method, and the resulting templates are stored in an inventory. For synthesis of a new utterance, given a phonetic transcription, the system uses rules to select the appropriate units, extracts them from the inventory and concatenates them.

IX. CONCATENATION

Now, the phonemes can be concatenated to give various words. Since, all these sounds (phonemes) are just column vectors, their constituent elements could be placed one after another and stored in another variable (vector). This is concatenation. This way all the words could be played by merely selecting the phonemes and placing the phoneme vectors one after another.

Sample Code:
```matlab
[y,fs1,ns1]=wavread('pain.wav');
yAIN=y(12000:1:16000);
[z,fs1,ns1]=wavread('gain.wav');
zG=z(150:1:950);
new=[zG;yAIN];
```

In this code, the sound ‘AIN’ from ‘pain’ is extracted and stored in yAIN and the sound ‘G’ from ‘gain’ is extracted and stored in zG. Therefore, to obtain the sound of ‘gain’ using these two variables we perform concatenation as shown above. Thus, after this stage a dictionary of words (with ‘.wav’ extension) has been created and stored for future reference. The phonemic representation of all words is also stored in this dictionary. For example, phonemic representation of ‘girl’ is g/ur/l. It helps in separating the phonemes of the whole word. The dictionary gets stored in the matlab directory as a text file.
Then the sounds pertaining to the constituent phonemes of the word Girl i.e. /g/,/ur/,/l/ were also extracted from different standard words.

![Spectrogram Of /g/](image1)

![Spectrogram Of /ur/](image2)

![Spectrogram Of /l/](image3)

After the three phonemes were extracted, now they are concatenated to construct back the word girl. The spectrogram of the resultant i.e. the concatenated “girl” is also displayed as follows:

![Spectrogram of Girl after concatenation](image4)

Both the Spectrograms i.e Girl (figure 6) and Girl (figure 8) are compared and similarities are found. The concatenated sound is much closer to the original. It is obvious that the degree of similarity increases with the precision in extracting the 44 phonemes which is done by trial and error method.

XII. CONCLUSION

This paper describes the successful implementation of a simple text to speech conversion by simple matrix operations. Hence this method is very easy and efficient to implement unlike other methods which involve many complex algorithms and techniques. The next step in improving this technique would be implementing some machine learning algorithms in order to support generalization.

ACKNOWLEDGMENT

We would like to acknowledge and extend our heartfelt gratitude to the following persons, the project team members, Ms Twinkle Tripathy, Mr. Gaurav Kanungo & Ms Anupama Minz for their vital support.

REFERENCES


