Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 1, 2004

Feature extraction and speech labeling for continuous speech recognition

Alexandru Căruntu[1], Gavril Toderean[1]

Abstract – Speech analysis and speech labeling are two important issues of any speech recognition system. Due to the fact that speech is analyzed on short – time intervals it is hard to make a visual interface for analysis. On the other hand it is known that speech recognition gives better results if signal's features are used for labeling. The program described in this paper is an attempt to realize an interface for speech analysis and a feasible framework for speech labeling. The application is developed in Visual C++ 6.0, and allows the extraction and visualization of speech features frame by frame and speech labeling at word and phone level.
Keywords: speech processing, feature extraction, speech analysis, speech labeling

I. INTRODUCTION

Speech signals are real, continuous, finite energy waveforms. Although they vary in time, on short periods (15 to 30 ms) they can be considered stationary and their properties can be analyzed. The analysis of the speech signal aims to determine a set of parameters which describes the important characteristics of speech. The fact that speech signals must be analyzed on short intervals makes very difficult the implementation of a visual interface. Most of the existing programs on this field extract the features using commands from the command line with many options hard to remember, and have a few or even none displaying tools. The program that we developed tries to overcome all these problems, using a visual interface which allows the visualization of the features of the speech signal frame by frame. Another important issue when dealing with speech recognition is speech labeling, that is defining word or phone boundaries. This paper is organized as follows: first the preprocessing, analysis and labeling of the speech signals are described, then are given some details about the implementation, and finally conclusions and remarks about the future directions to follow are presented.

II. SPEECH SIGNAL PREPROCESSING

First some *preprocessing* is applied to speech wave. Because most part of the energy of the signal lies between 50 Hz and 4 KHz a low – pass or a band – pass *filtering* is required. This way, low – pass components, which do not contain useful information, are eliminated. The upper constraint is necessary in order to avoid the aliasing which appears thru sampling. Next, the speech signal is *digitized* with the help of an analog – to - digital converter, with a resolution between 8 and 16 bits. To eliminate the effects of high frequencies attenuation, speech signal is *pre – emphasized*. Last step before analysis is *segmentation*, which is realized usually with a Hamming window (Figure 1). For better results frame overlapping is recommended.

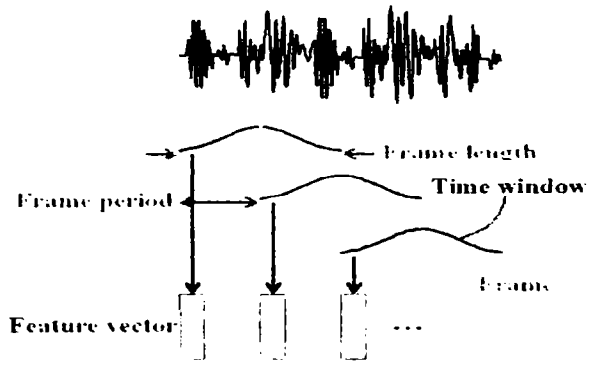

Fig. 1. Speech segmentation [1].

III. SPEECH ANALYSIS

By analyzing the signal in time domain we obtain the maximum and medium amplitude, energy, zero – crossing rate and fundamental frequency. Maximum amplitude gives us information about the voiced or unvoiced character of the speech frame. Time interval between two successive maximums corresponds to fundamental period.

Short – time energy of speech wave is defined as:

$$E(n) = \frac{1}{N}\sum_{m=0}^{N-1}[w(m)s(n-m)]^2, \qquad (1)$$

[1] Technical University of Cluj-Napoca,
e-mail: Alexandru.Mihai.Caruntu@com.utcluj.ro

where w(m) is the window, N number of samples and s(n-m) is a sample from the signal. This parameter gives us information about the voiced (high energy) or unvoiced (low energy) nature of the speech frame. Also, it is very useful in the silence detection in isolated words recognition process.

Zero – crossing rate is calculated with the formula:

$$ZCR=\sum_{n=0}^{N-2}\frac{1-\mathrm{sgn}[s(n)]\mathrm{sgn}[s(n+1)]}{2}. \quad (2)$$

This parameter estimates the fundamental frequency of the speech wave. Also, together with the energy, helps to silence detection.

One of the methods, which are used to determine the fundamental frequency, is based on the autocorrelation function defined as:

$$R_n(k)=\sum_{m=-\infty}^{\infty}s(m)w(n-m)s(m-k)w(n-m+k). \quad (3)$$

For periodical signals the *autocorrelation* function has maximums at regular intervals, aspect which is used to determine the fundamental frequency [2].

Frequency domain analysis gives better features for processing than time domain analysis. The excitation and vocal tract can be easily separated in spectral domain. While different utterances of the same sentence can differ in time domain, in frequency domain they are similar. Also, human ear is more sensitive to aspects related to the amplitude of the speech signal than related to its phase, so spectral analysis is used most of the time to extract features that describe speech signal.

The most common method to analyze speech in frequency domain is *Fast Fourier Transform*. In time, a considerable number of algorithms that exploit the advantages of modern processors have been developed. In our implementation we used a radix-2 one.

Linear Predictive Coding analysis provides a good p g y automatic speech recognition systems. This method fits the parameters of an all – ole model to the speech spectrum, although the spectrum itself is not computed explicitly. LPC coefficients can be found using either autocorrelation method, either covariance method. In our implementation we choose the first one because of its popularity and because it gives also the *reflection (or PARCOR) coefficients*. The autocorrelation function is evaluated first and the results are converted to LPC coefficients using Levinson Durbin algorithm:

$$E^0 = R(0)$$

$$k_i=\left(R(i)-\sum_{j=1}^{i-1}\alpha_j^{(i-1)}\cdot R(i-j)\right)\Big/E^{(i-1)},\ i=\overline{1,p}$$

$$\begin{aligned}\alpha_i^{(i)} &= k_i \\ \alpha_j^{(i)} &= \alpha_j^{(i-1)}-k_i\cdot\alpha_{i-j}^{(i-1)} \\ E^{(i)} &= (1-k_i^2)\cdot E^{(i-1)}\end{aligned} \quad (4)$$

where $\alpha_j=\alpha_j^{(p)}$, $1\le j\le p$ are the LPC coefficients, k_i are PARCOR coefficients and E is frame energy.

Cepstrum is defined as the IFFT of the logarithm of the spectrum. This analysis allows separation of the contribution of the source from that of the vocal tract in the speech signal. A set of features used widely in speech representation is that of cepstrum coefficients obtained from LPC coefficients:

$$c_n=\begin{cases}0 & n<0\\ \ln G & n=0\\ a_n+\sum_{k=1}^{n-1}\left(\frac{k}{n}\right)c_k a_{n-k} & 0<n\le p\\ \sum_{k=n-p}^{n-1}\left(\frac{k}{n}\right)c_k a_{n-k} & n>p\end{cases}. \quad (5)$$

This way the variability introduced by the excitation is eliminated and better results are obtained in recognition.

Mel cepstral analysis is a perceptual analysis which uses the Mel scale and a cepstral smoothing in order to obtain the final spectrum [3]. First the short – term spectrum of the speech frame is evaluated and then integrated over gradually widening frequency intervals on the Mel scale, with a triangular filter – bank as the one showed in Figure 2.

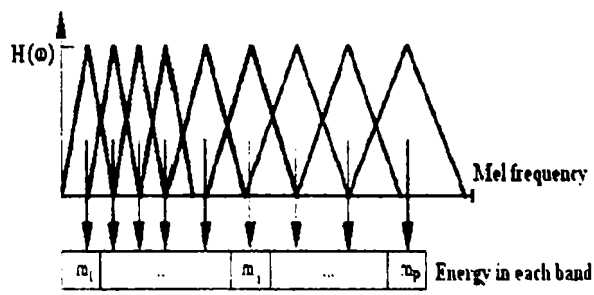

Fig. 2. Mel scale triangular filter – bank.

The filter – bank is given by:

$$H_m(\omega)=\begin{cases}\dfrac{\omega-\omega_{m-1}}{\omega_m-\omega_{m-1}}, & \omega_{m-1}\le\omega\le\omega_m\\ \dfrac{\omega_{m+1}-\omega}{\omega_{m+1}-\omega_m}, & \omega_m\le\omega\le\omega_{m+1}\\ 0, & otherwise\end{cases}, \quad (6)$$

with central frequencies:

$$\omega_m = \begin{cases} 100 \cdot m, & 0 \le m \le 10 \\ 1000 \cdot 1{,}15^{m-10}, & m > 10 \end{cases} \quad (7)$$

Next a vector with log energies is evaluated for each filter:

$$S_m = \ln\left[\sum_{k=0}^{N-1} |X(k)|^2 H_m(k)\right], \quad (8)$$

then the MFCC coefficients are obtained using a Discrete Cosine Transform:

$$c_n = \sum_{m=0}^{M-1} S_m \cos\left(\frac{\pi n(m+0{,}5)}{M}\right). \quad (9)$$

This transform is used because the coefficients obtained after the calculus of the power spectra are highly correlated and the cepstral coefficients are uncorrelated, fact that allows the number of parameters to be reduced. In practice they are using 24 to 40 filters but only the first 13 coefficients are used for speech recognition tasks.

IV. SPEECH LABELING

A very important issue when performing speech recognition is *speech labeling*, which is the process of finding word or phone boundaries. Very useful tools for this are *spectrograms*. A spectrogram of a time signal is a two-dimensional representation that displays time in its horizontal axis and frequency in its vertical axis. A gray scale is typically used to indicate the energy at each point with white representing low energy and black high energy [4]. The basic tool with which to compute them is short-time Fourier analysis.

There are two main types of spectrograms: *narrow-band* and *wide-band*. Wide-band spectrograms use relatively short windows (< 10 ms) and thus have good time resolution at the expense of lower frequency resolution, since the corresponding filters have wide bandwidths (> 200 Hz) and the harmonics cannot be seen. Narrow-band spectrograms use relatively long windows (> 20 ms), which lead to filters with narrow bandwidth (< 100 Hz). On the other hand, time resolution is lower than for wide-band spectrograms. In this case the harmonics can be clearly seen, since some of the filters capture the energy of the signal's harmonics and filters in between have little energy.

Spectrograms can aid in determining formant frequencies and fundamental frequency, as well as voiced and unvoiced regions.

The *transcription* can be done in several ways using different speech units, from words, if the speech recognition system is designed for a small word vocabulary, to phonemes, if we want to implement a large vocabulary continuous speech recognition system.

A number of *systems for the phonetic transcription* of speech are used, among them IPA and Worldbet, but we can define also our own phonetic system.

Margins of the phones can be marked in many ways. We mention here only two of them: using samples as the ones who designated TIMIT database did, or time moments as the HTK program does.

V. IMPLEMENTATION

The program is a MDI application developed in Visual C++ called Visual Speech Analyzer (VSA). It has three main modules: sound recording, speech analysis and speech labeling.

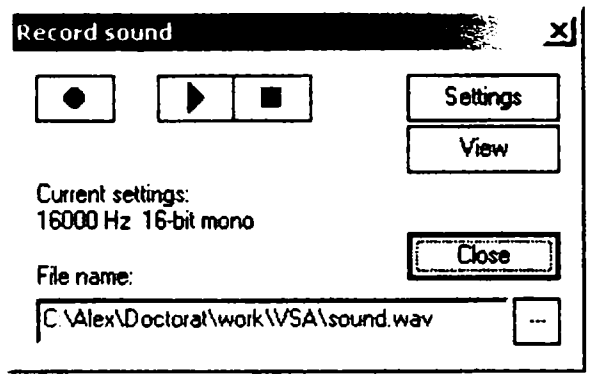

Fig. 3. The sound recorder.

The *sound recorder* (Fig. 3) allows recording and playing of a sound data. The supported settings can be seen in Fig. 4. If the button view is pressed the waveform corresponding to recorded sound is displayed.

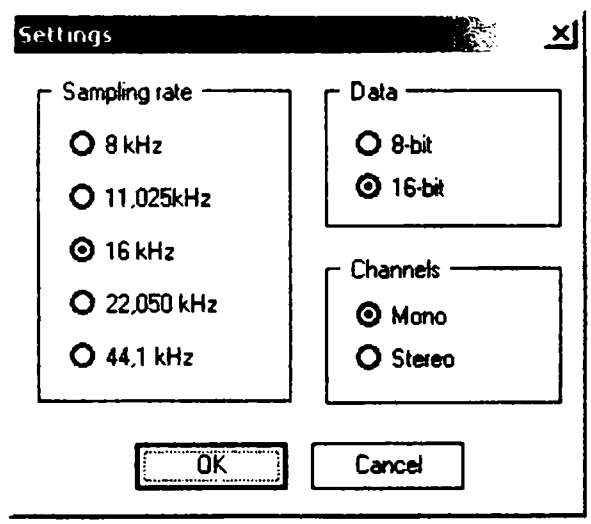

Fig. 4. The settings of the sound recorder.

The *analysis module* implements the methods described in the third paragraph of this paper.

The interface is presented in Fig. 5.

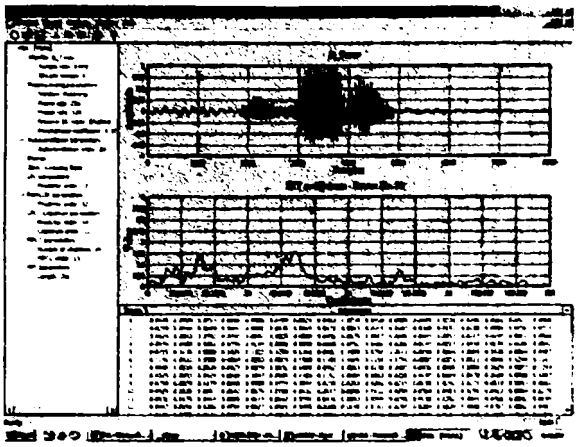

Fig. 5. The interface for analysis.

The Project option from the menu allows the user to create a new project for speech analysis, open an existing one or save the current one. We can open a wav file for analysis with the command Open from the menu Sound. Also the preprocessing described on section 2 is implemented here. With the command Preprocessing parameters the user can select the window applied to the signal (Hamming or rectangular), the frame size and frame rate (both of them expressed in number of samples), whether to remove or not the DC mean or pre - emphasize the signal. The analysis methods described in section 3 can be found under the menu option Analysis. Their results can be saved in a file if the user validates this option.

The interface consists from a tree control which displays information about the wave that is analyzed, the preprocessing parameters and the parameters of each analysis that has been selected. The sound that is analyzed is shown on the upper graph while the features values are shown on the second one. They are displayed frame by frame as the user moves with the mouse on the graph of the signal. Finally, there is a list control which displays features values for all the frames of the signal. The user can switch between parameters by double – clicking on the parameters name from the tree control [5].

The *labeling module* displays the waveform and the corresponding narrow-band and wide-band spectrograms. Instead of time we used number of samples as measurement unit for x axis. The length of the window is calculated as the next power of two greater than the number of samples corresponding for 20 ms of speech for narrow – band spectrograms and the last power of two before the number of samples corresponding for 20 ms of speech for wide-band spectrograms. Two cursors are available for marking different possitions on the speech wave.

On the right side of the window are two lists which displays the starting and ending sample for each word or phonema.

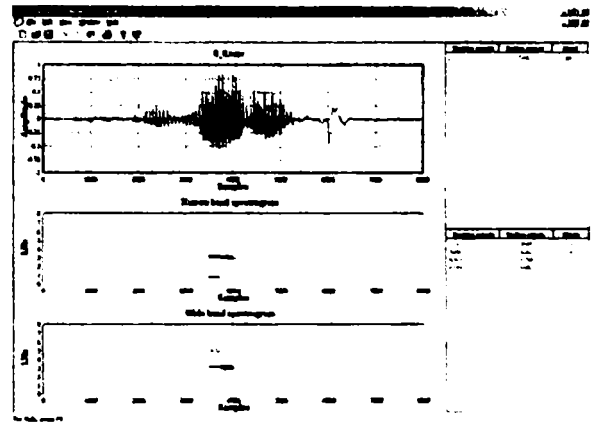

Fig. 6. The interface for labeling.

VI. CONCLUSIONS AND FUTURE WORK

The application described in this paper allows the features extraction of the speech signal and speech labeling. We intend to use the parameters extracted this way for an application that performs continuous speech recognition.

REFERENCES

[1] S. Furui, *State of the Art Speech Recognition Technology*.

[2] A. Căruntu, *Stadiul actual în domeniul recunoaşterii vorbirii continue*, Referat I, 2003.

[3] P.G. Pop, G. Toderean,"Comparison of Feature Parameters Used In Speaker Recognition", *Acta Tehnica Napocensis*, Vol. 43, No. 2, p. 43 - 46.

[4] X. Huang, A. Acero, H-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice Hall, 2000.

[5] A. Căruntu, G. Toderean, "Feature Extraction for Continuous Speech Recognition", *A&QT-R 2004 (THETA 14) IEEE-TTTC - International Conference on Automation, Quality and Testing, Robotics*, May 13 - 15, 2004, Cluj-Napoca, Romania.