SOFTWARE FOR SPEECH ANALYSIS

Alexandru Caruntu Gavril Toderean Alina Nica
e-mail: Alexandru.Caruntu@com.utcluj.ro e-mail:Gavril.Toderean@com.utcluj.ro e-mail:Alina.Nica@com.utcluj.ro
Technical University of Cluj-Napoca, Faculty of Electronics and Telecommunications, Department of Telecommunications, 26, Baritiu str., Cluj-Napoca, Romania

Key words: Speech processing, feature extraction, speech analysis

ABSTRACT

Creating a visual interface for speech analysis is not an easy task due to the fact that speech must be analyzed on short-time intervals. We present here a tool which contains some of the fundamental speech analysis techniques implemented in a visual manner.

I. INTRODUCTION

Speech analysis is the first step in any application that involves speech, whether is a speech recognition system or a text-to-speech one. It is also the starting point in the learning process for every student. With these facts in mind, we tried to design a speech analysis tool which can be helpful for both sides.

Speech signals are real, continuous, finite energy waveforms. Although they vary in time, on short periods (10 to 30 ms) they can be considered stationary and their properties can be analysed, aiming to determine a set of parameters which describes the important characteristics of speech. The need for a short-time analysis 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 [1 – 7]. We developed a software which:

- Allows the analysis of a speech signal frame by frame.
- It is capable to represent various analysis methods in time and frequency domain.
- Can display the values of the features obtained as a result of a certain analysis method.
- Allow the use of obtained features by another application.

This paper is organized as follows: first the preprocessing and analysis of the speech signals are described, emphasizing the manner in which we have implemented them. A small example, conclusions and future plans will finish the paper.

II. SPEECH SIGNAL PREPROCESSING

Before starting the analysis, the speech wave needs to be preprocessed [8 - 11]. 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 analogue–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. For better results frame overlapping is recommended.

III. SPEECH ANALYSIS

Many analysis techniques have been developed in time. For this release we choose the most used of them. Short description and details of their implementation will be given in this paragraph.

TIME DOMAIN ANALYSIS

Autocorrelation, energy and zero-crossing rate are estimated in time domain.

Autocorrelation is one of the methods used to estimate the *fundamental frequency* of speech signal. The autocorrelation function is defined as [12]

$$R_n(k) = \sum_{m=-\infty}^{\infty} s(m)w(n-m)s(m-k)w(n-m+k), \quad (1)$$

where *s(m)* is a sample from speech signal and *w(m)* is the window (usually a rectangular window it is used).

For periodic signals the autocorrelation function has maximums at regular intervals, aspect which is used to determine the fundamental frequency. Our toolbox displays the autocorrelation function for a certain frame as well as the values for autocorrelation coefficients for all the frames in the speech signal.

Short – time energy of the speech wave is defined as [12]:

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

where N is the number of samples. A frame with high energy corresponds to a voiced portion of speech signal, while one with low energy to an unvoiced region.

Zero – crossing rate is calculated with the formula [12]:

$$ZCR = \sum_{n=0}^{N-2}\frac{1-\mathrm{sgn}[s(n)]\mathrm{sgn}[s(n+1)]}{2}. \qquad (3)$$

This parameter estimates the fundamental frequency of the speech wave. For example, for a sine wave of frequency f_0 , ZCR is $2f_0$ [12].

With our software the user can see the evolution in time for each of these two parameters as well as their values for every frame of the speech signal.

SHORT-TINE FOURIER ANALYSIS

Frequency domain analysis provides better features to represent the speech signal than those from time domain. 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.

Numerous algorithms that exploit the advantages of modern processors have been developed in order to obtain the *Fourier transform* of the signal. In our implementation we used a radix-2 algorithm with decimation in time based on Danielson-Lanczos Lemma [13]. The spectrum of the signal is calculated and displayed on a logarithmic scale together with spectral coefficients values.

LINEAR PREDICTIVE CODING

Linear Predictive Coding is one of the most powerful speech analysis techniques. The basic idea of this method is that a speech sample can be approximated as a linear combination of past speech samples [12]. The result of this technique is the LPC coefficients, which can be found using either autocorrelation method, either covariance method. In our implementation we chose the first one because it gives also the *reflection* and *PARCOR coefficients*.

The autocorrelation function is evaluated first and the results are converted to LPC coefficients using Levinson-Durbin algorithm [14]:

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

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

$$\alpha_i^{(i)} = k_i \qquad (4)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \cdot \alpha_{i-j}^{(i-1)}$$

$$E^{(i)} = (1-k_i^2) \cdot E^{(i-1)}$$

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

For this method we display the logarithmic spectrum and the values of LPC coefficients frame by frame. For obtaining the spectrum, LPC coefficients are padded with zeroes until FFT length is reached, than an FFT is applied to the vector obtained in this manner [11], [15].

CEPSTRAL ANALYSIS

Another set of features used to represent speech signal can be obtained through cepstral analysis. This analysis allows separation of the contribution of the source from that of the vocal tract in the speech signal.

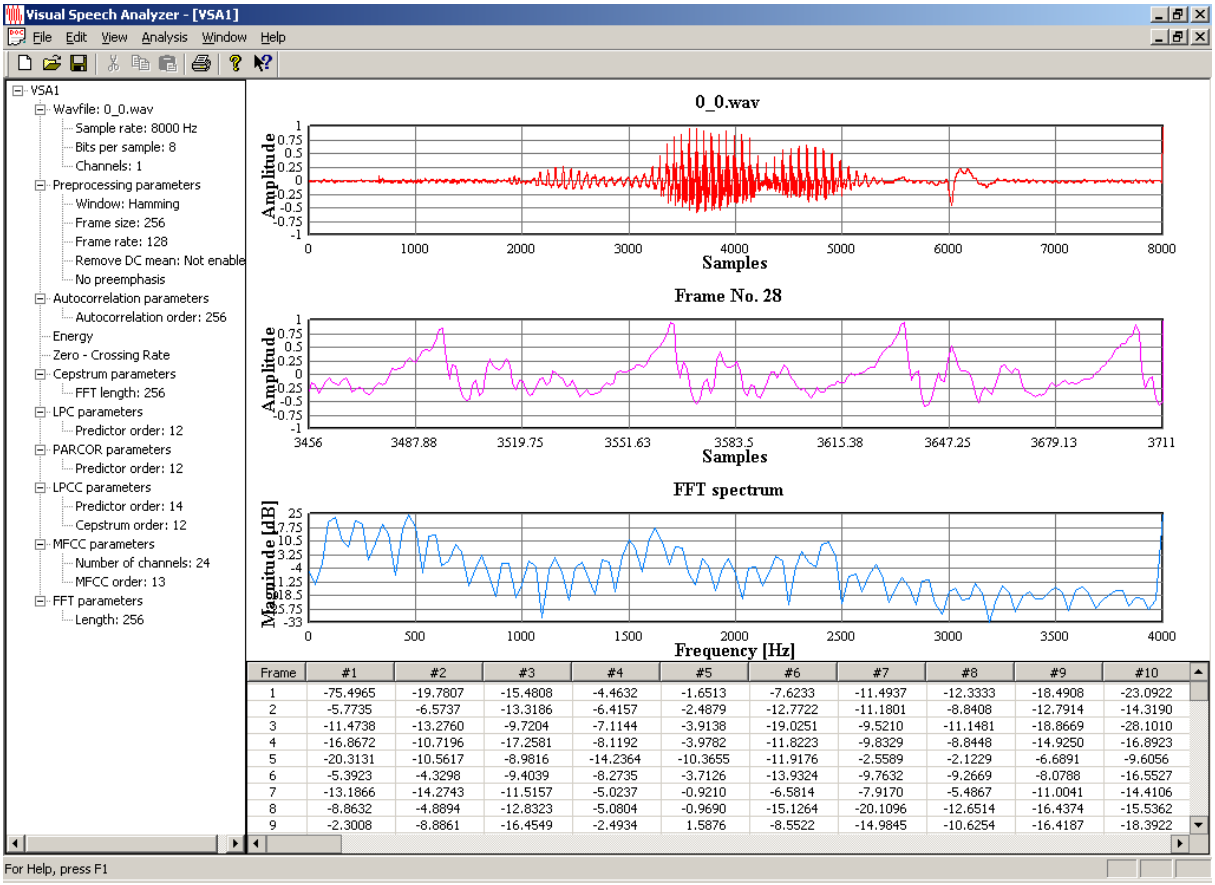
Cepstrum is defined as the IFFT of the logarithm of the spectrum. Usually, cepstral coefficients are calculated from LPC coefficients, in an attempt to sum up the advantages of both techniques [8]:

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

A representation of short-term real cepstrum in quefrency domain can be obtained using our software. The LPCC spectrum can be visualized also on a logarithmic scale. The values of the cepstral coefficients for all the frames in the speech signal are displayed too.

MEL-CEPSTRAL ANALYSIS

Mel-cepstral analysis is the technique which provides the best set of features for speech recognition systems. In order to obtain the spectrum, Mel scale and a cepstral smoothing are used. Our implementation is based on [16].



Frame	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
1	-75.4965	-19.7807	-15.4808	-4.4632	-1.6513	-7.6233	-11.4937	-12.3333	-18.4908	-23.0922
2	-5.7735	-6.5737	-13.3186	-6.4157	-2.4879	-12.7722	-11.1801	-8.8408	-12.7914	-14.3190
3	-11.4738	-13.2760	-9.7204	-7.1144	-3.9138	-19.0251	-9.5210	-11.1481	-18.8669	-28.1010
4	-16.8672	-10.7196	-17.2581	-8.1192	-3.9782	-11.8223	-9.8329	-8.8448	-14.9250	-16.8923
5	-20.3131	-10.5617	-8.9816	-14.2364	-10.3655	-11.9176	-2.5589	-2.1229	-6.6891	-9.6056
6	-5.3923	-4.3298	-9.4039	-8.2735	-3.7126	-13.9324	-9.7632	-9.2669	-8.0788	-16.5527
7	-13.1866	-14.2743	-11.5157	-5.0237	-0.9210	-6.5814	-7.9170	-5.4867	-11.0041	-14.4106
8	-8.8632	-4.8894	-12.8323	-5.0804	-0.9690	-15.1264	-20.1096	-12.6514	-16.4374	-15.5362
9	-2.3008	-8.8861	-16.4549	-2.4934	1.5876	-8.5522	-14.9845	-10.6254	-16.4187	-18.3922

Figure 1. The interface of *Visual Speech Analyzer*

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.

The bandwidths of the filter–bank are given by:

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

while for the central frequencies calculus we used the approximation:

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

Next step is the evaluation of the energy logarithm for each filter:

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

followed by the computation of MFCC coefficients, using a Discrete Cosine Transform:

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

Spectrum and MFCC coefficients values can be obtained for this technique too.

IV. RESULTS AND DISCUSSION

Our application was developed in Visual C++ 6.0. We named it Visual Speech Analyzer and its interface can be seen in Figure 1. Each of the analysis techniques described in previous section can be found under option *Analysis* from the menu.

After selecting an analysis method its parameters must be set (e.g. for an FFT analysis the user must provide its length in samples). Three graphs are used to display

the speech signal to be analyzed, current frame, and analysis result, which can be displayed in time or frequency domain as shown in previous section. Each frame is selected by moving the mouse over the speech signal. As the mouse moves, the graphs reflecting the current frame and analysis result are changing, showing the new values for the new position of the mouse.

At the bottom of the window the values for the features obtained thru analysis can be seen.

On the left we find details about the speech wave currently analyzed (sample rate, bits per sample, channels) as well as the preprocessing parameters for the current analysis (what window we used, its size in samples, the overlap between successive frames, also in samples, if DC mean was removed or preemphasis was applied).

A history of the analysis performed for the current wave, preprocessed with certain parameters, is also provided on the left side of the window. If user wants to review those analyses, it can simply select them from the list.

All the features obtained can be saved in a file if the user enables this option. The first two values saved are the number of frames and the number of features in a frame, followed by the values of the features. This way, any application developed in any programming language can use them as input values.

A sound recording and playing module is also implemented, supporting currently only the WAVE format.

V. CONCLUSION

We presented here a new toolbox for speech analysis as an alternative to existing publicly available software in this field. A number of fundamental analyses were implemented to sustain our approach. We developed a tool suited very well to learning, but also useful to those who want to build speech recognition or text-to-speech systems.

We intend to make from *Visual Speech Analyzer* a long-term project. New analysis will be added, pursuing an educational goal, as well as a speech labelling tool.

REFERENCES

1. *** Speech Filling System, http://www.phon.ucl.ac.uk/resource/sfs/
2. *** Praat, http://www.fon.hum.uva.nl/praat/
3. *** Speech Analyzer, http://www.sil.org/computing/speechtools/
4. *** Voicebox, http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html
5. *** Auditory Toolbox, http://rvl4.ecn.purdue.edu/~malcolm/interval/1998-010/
6. *** Colea, http://www.utdallas.edu/~loizou/speech
7. *** HTK, htk.eng.cam.ac.uk/
8. Furui, S., Digital Speech Processing, Synthesis and Recognition, Second Edition, Revised and Expanded, Marcel Dekker, Inc. 2001.
9. Holmes, J., Holmes, W., Speech Synthesis and Recognition, Second Edition, Taylor&Francis, 2001.
10. O'Shaughnessy, D., Speech Communications: Human and Machine, Second Edition, IEEE Press, 2000.
11. Picone, J., "Signal Modelling Techniques in Speech Recognition", IEEE Proceedings, Vol. 81, No. 9, pp. 1215-1247, 1993.
12. Rabiner, L. R., Schafer, R. W., Digital Processing of Speech Signals, Prentice-Hall, Englewood Cliffs, 1978.
13. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., Numerical Recipes in C: The Art of Scientific Computing, Second Edition, Cambridge University Press, 1992.
14. Young, S., Evermann, G., Kershaw, D, Moore, G, Odell, J, Ollason, D., Povey, D., Valtchev, V., Woodland, P, The HTK Book, Cambridge University Engineering Department, 2002.
15. Deller Jr., J.R., Hansen, J.H.L., Proakis, J.G., Discrete-Time Processing of Speech Signals, IEEE Press, 2000.
16. Huang, X., Acero, A., Hon, H., Spoken Language Processing: A Guide to Theory, Algorithm, and System Development, Prentice-Hall, 2001.