COMPARISON OF ENERGY-BASED ENDPOINT DETECTORS FOR SPEECH SIGNAL PROCESSING

A. Ganapathiraju, L. Webster, J. Trimble, K. Bush, P. Kornman

Department of Electrical and Computer Engineering
Mississippi State University,
Mississippi State, Mississippi -39762

***Abstract* - Accurate endpoint detection is a necessary capability for construction of speech databases from field recordings. In this paper we describe the implementation of two endpoint detection algorithms which use signal features based on energy and rate of zero crossings. We have made extensive use of object-oriented concepts and data-driven programming to make our code re-usable for a variety of applications, including speech recognition. A uniform user-interface for both algorithms has been developed using a novel virtual class methodology.We also present a comparison of the two algorithms using an objective evaluation paradigm we have developed. A small locally prepared database has been used for the purpose of evaluation.**

INTRODUCTION

The accurate determination of speech in the presence of background noise is very important in many areas of speech processing. In the early stages of speech research, endpoint detectors were an integral part of the isolated word speech recognizer to be used for the alignment process. With the advent of continuous speech recognizers, alignment is now done automatically [1]. Endpoint detectors have found extensive usage as data trimming tools in the preparation of speech databases.

With increasing vocabulary sizes it is imperative to train recognizers on large sets of data. Collecting speech data over telephone has been found to be the best way to develop such databases. This data has to be efficiently segmented before recognition development can commence. Speech endpoint detectors are used for this purpose. However, endpoint detection is not a trivial process. In such high-noise conditions the energy levels of speech and background noise are not very different. In high-noise environments, the characteristics of the speech signal are very similar to those of the channel and speech discrimination from the background noise becomes a difficult problem. For example, data recorded over telephone lines can exhibit an SNR as low as 5 dB for certain segments of speech such as fricatives and nasals [6].

Various algorithms proposed for speech endpoint detection [2-5, 9] differ in many respects such as the features of the speech signal employed (e.g. energy, spectral slope, periodicity, zero crossing rate), complexity and even the applications for which they are intended. The most popular, flexible, and accurate algorithms use signal energy as the feature for word boundary detection.

In this paper we present the implementation of two energy-based algorithms. We have emphasized the need to make the system extensible and the code reusable in the development of the relevant software. The paper will also discuss the algorithms and their implementation along with the structure of the software. Finally a detailed comparison of the two algorithms will also be presented.

ALGORITHMS

Energy has been used as the feature for detecting endpoints in speech utterances since the 1970s. The popularity of energy based algorithms is attributed to the fact that computing energy from the speech signal is a simple operation compared to extracting other features. By incorporating good smoothing functions in algorithms the detection process can be made very accurate. The energy contour of the speech signal has been found to be a good indicator of presence of speech in signals with a moderate SNR.

ENERGY ALGORITHM:

When digitized speech is input to the system, the data is preemphasized with a simple first order low-pass filter. A preemphasis factor of 0.95 is typically used in speech recognition applications; and for the purpose of compatibility with other systems, we choose the same value. The basic feature extraction is done next by computing the short time energy of the speech signal. Speech can be safely assumed to be stationary in an interval of 10 ms; a common frame duration is 20 ms. The frame window has an overlap so that there is a smooth transition of the features from frame to frame.

Initially the various parameters like nominal noise energy and nominal signal energy are set to default values. The thresholds are dynamically updated depending on the current observation frame. The use of a circular buffer for processing the input speech allows us to make use of a

latent delay to update parameters while processing a frame of data. The problem is then converted into a problem of identifying a particular pattern in a state machine.

Depending on the energy levels of the speech signal and the background noise, the frame-level outputs of the energy calculation are converted into a state machine representation that uses four states. Fig. 1 illustrates the design of this state machine. Many smoothing operations are performed on the state machine depending on the user-supplied information on the minimum utterance duration (MUD) and the minimum utterance separation (MUS). Smoothing operations are required not only to make the detection efficient but also to make efficient use of computer memory. Fig. 2 demonstrates an example of the state machine output for a simple case of a string of isolated word utterances. Note that though the signal energy rises beyond the threshold for a short time at the beginning this occurrence does not meet the MUD constraint and will not be treated as a beginning of speech.

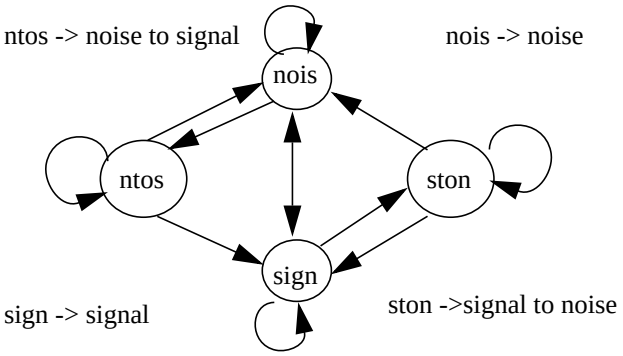

Figure 1. Generation of the State Machine

Long stretches of silence can be deleted from the circular buffer once it has been confirmed that the minimum utterance separation constraint has been met. Long stretches of signal can be removed once the minimum utterance duration constraint is met. Some other smoothing functions remove short bursts of noise if duration constraints are not met. The state machine output is then analyzed to output the endpoint information. Beginning of speech is declared if a stretch of 'sign' state occurs for a duration greater than the minimum utterance duration. End of an utterance is declared if a stretch of 'nois' state occurs for a duration greater than the minimum utterance separation.

ENERGY AND ZERO CROSSING ALGORITHM:

The energy and zero crossing algorithm is an extension of the energy algorithm. At the stage in the energy algorithm where the energy is calculated the zero crossing rate is also computed [6]. The zero crossing rate information can be used to refine the endpoints found by using just the energy information. The situations where using only energy information could fail in determining accurate endpoints are often weak fricatives (/f/, /h/), nasals at end ("gone") and the trailing vowels at the end ("zoo") [7]. In each of the above cases there will be a certain amount of zero crossing activity in these regions where the energy levels fall below a level where detection becomes inaccurate. Zero crossing activity is sought in an interval of about 200 ms preceding the initial guess about the begin point. If more than 120 ms have such activity above a certain threshold, the begin point is moved to the first frame where the zero crossing rate went beyond the threshold [6]. A similar procedure is followed for the end of utterances.

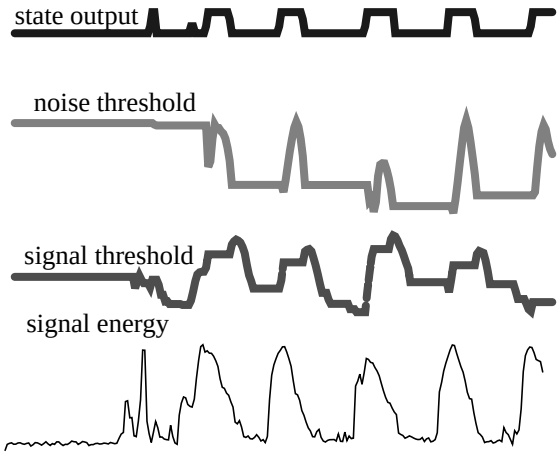

Figure 2. Algorithm Overview

SOFTWARE

The algorithms are implemented in a structured software making extensive use of object-oriented concepts and ideas of data-driven programming. The main aim of the software is to give the user full control over the choice of the algorithm he/she wishes to apply and also the parameters which go into the algorithm. For this purpose a uniform user-interface is required. The solution to this problem is to employ object-oriented programming, particularly the virtual functions concept. Figure 3 gives a block diagram representation of the software structure. The user can specify the parameters and the algorithm to use. The most important feature of C++ we use in the software is the virtual function mechanism. In many situations it is desirable to hide the implementation specifics from the user.

For example, in endpoint detection we should have the

flexibility to use any specified algorithm without changing the software. In the virtual-function approach, the parent object class 'Signal_detector' is created that contains a standard set of virtual functions that represent the software interface to the user. At run-time when a particular algorithm is requested, the virtual functions are overloaded with the specific algorithm properties.

The parameter file given by the user is used to decide on which algorithm is to be implemented. It also contains the sample frequency, frame duration and the energy thresholds. At run-time the object representing the specific algorithm is created transparently and then the overloaded virtual functions come into play. The system is easily extendable with this structure to more algorithms. Though this methodology could have been implemented using ANSI C, the code gets too complex and memory cannot be used efficiently. With C++ the necessary features are built into the language. The code is built in a modular fashion to allow for any modifications to the existing functions without having to access a very large chunk of code.

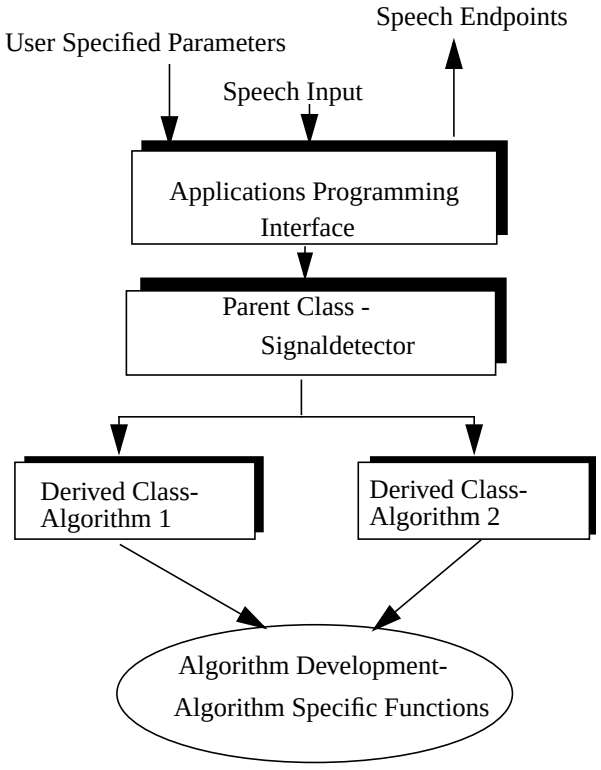

Figure 3. Software Structure

The use of a circular buffer to read in the speech data in a frame-by-frame basis helps in two important ways. It makes the code capable of performing real-time processing and it helps simplify the smoothing and filtering operations to be performed. Various levels of debugging are allowed by the code. The user can get information on the energy levels in each frame of analysis or information before and after the smoothing processes.

EXPERIMENTS

DATABASE:

The evaluation database was recorded under typical office environment conditions. The data was recorded on two 16 kHz channels and stored as 16 bit integers. Two sets of microphones and pre-amps were used in the recording of the data. The database is composed of three male and three female speakers with unique voice characteristics. The types of utterances recorded are shown in Table 1. The typical length of each recording was 20 secs.

OBJECTIVE EVALUATION:

An objective evaluation paradigm was developed for the comparison of the two algorithms. The performance of both the algorithms was analyzed using the same hand-marked database. A scoring function was defined to give an objective measure of the errors by the two algorithms.

It is a well known fact that humans can tolerate a certain amount of error in recognition of endpoints of utterances. Beyond a certain threshold the acceptability of the error decreases steeply. Also, we tend to heavily penalize completely missing an utterance or splitting a word in parts. A nonlinear scoring function was used to implement these constraints. The system is not given any penalty for an error in finding the endpoints less than 5 ms from the true endpoints. Errors beyond this point accrue a penalty on a linear scale until a maximum penalty of 1 is reached for errors greater than or equal to 0.5 secs. Additions and deletions of endpoints are also given a similarly strong penalty of 1 each.

Algorithm	Sub	Del	Ins	Avg
Energy	0.23	2.0	1.0	1.10
Energy and Zerocrossing	0.24	2.0	1.0	1.06

Table 1: Comparison of algorithms

The number of errors can be very large in case the parameters are not chosen judiciously. It has been found during the many test runs of our system that the minimum utterance separation and the minimum utterance duration play a vital part in the performance of the algorithms. These need to be varied according to the type of utterance. It was not possible to decide on one set of parameters for

all types of utterances without a large error rate. Table 2 gives the values we used for MUD and MUS.

RESULTS

The experiments were iterated a number of times before we could get the present set of optimized values. Table 1 and Table 2 give the results of our experiments and form a basis for the comparison of the two algorithms.We can infer that the overall performance of the algorithms is comparable. This is due to the fact that the database was dominated by medium SNR conditions. The comparative performances of the algorithms on individual types of utterances shows that using zero crossing information along with energy will improve the performance of the endpoint detector. The effect of using different windows on the performance of the algorithms has to be studied. The algorithms need to be run on a more comprehensive database representative of a wide range of SNR.

Utterance	E	E&ZC	MUD (secs)	MUS (secs)
Isolated Digits	0.11	0.12	020	0.06
Teens	3.10	3.06	0.40	0.10
Multi-syl Dig.	0.12	0.11	0.30	0.10
Long Dig. Str.	1.17	1.15	1.0	0.10
Phrases	0.24	0.28	1.0	0.10
Sentences	0.20	0.32	1.0	0.10
Spont. Speech	0.73	0.71	3.0	0.10

Table 2: Performance by Utterance type

CONCLUSIONS

This paper introduces a novel software implementation of two energy-based algorithms and compares the performance of both. No significant improvement was observed by using zero crossing information in conjunction with signal energy.

The software structure has been designed so as to make it reusable for a variety of applications. We intend to use the same structure in building the front-end of a speech recognition system. It will also be used for the search and alignment algorithms in the speech recognizer where different algorithms have to be tried without affecting the overall software structure. With the extensive use of more complex algorithms using Hidden Markov Models and Artificial Neural Networks the application of endpoint detectors in recognition has gradually diminished; though the need for large databases for training algorithms has increased for large vocabulary applications. The traditional energy-based endpoint detectors are now important tools in the automatic segmentation of these databases.

The software for the endpoint detector and all supporting tools are available via anonymous ftp to isip.msstate.edu.

ACKNOWLEDMENTS

The authors are grateful to Dr. Joseph Picone for all the support he has given towards this project and also supplying us with software tools to perform this project. We would also like to thank Dr. William Ebel for his insightful suggestions. We would like to acknowledge the work done by late Paul Kornman towards this project. His contribution has been instrumental in making this project a reality.

REFERENCES

1. B.Wheatley, et. al.,“Robust Automatic Time Alignment of Orthographic Transcriptions with Unconstrained Speech,” *Proc. ICASSP 92,* Vol. 1, pp. 533-536, 1992.
2. J. Junqua et. al.,“A Robust Algorithm for Word Boundary Detection in Presence of Noise,”*IEEE Trans. on Speech and Audio Processing*, Vol. 2, No. 3, pp. 406-412, July 1994.
3. L.F. Lamel et. al., “An Improved Endpoint Detector for Isolated Word Recognition,” *IEEE Trans. Acoustics., Speech, Signal Processing*, Vol. 29 pp. 777-785, Aug. 1981.
4. L.R. Rabiner and M.R. Sambur, “An algorithm for determining the endpoints of isolated utterances,” *Bell Syst. Tech. J.*, Vol. 54, pp. 297-315, 1975.
5. Minsoo Hahn et. al.,” An Improved Speech Detection Algorithm for Isolated Korean Utterances,”*Proc. ICASSP 92*, Vol. 1, pp. 525-528,1992.
6. L.R. Rabiner and R.W. Schafer,”*Digital Processing of Speech Signals*,” Englewood Cliffs,N.J., Prentice-Hall,1978.
7. J.R.Deller et. al.,” *Discrete Time Processing of Speech Signals*,” N.Y., Macmillan, 1993.
8. B. Reaves,“Comments on an improved endpoint detector for isolated word recognition,” *Corresp. IEEE Acoust.,Speech and Signal Processing*, Vol.39, pp. 526-527, Feb.1991.
9. R.Tucker, “Voice Activity Detection Using a Periodicity Measure,” *IEE Proceedings, Part 1, Communications, Speech, Vision*, Vol. 139, pp. 377-380, Aug. 1992.