ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/3333812

Signal modeling for high-performance robust isolated word recognition

Article *in* IEEE Transactions on Speech and Audio Processing · October 2001

DOI: 10.1109/89.943342 · Source: IEEE Xplore

CITATIONS
33

READS
104

2 authors:

Montri Karnjanadecha
Prince of Songkla University
41 PUBLICATIONS **235** CITATIONS

SEE PROFILE

Stephen A. Zahorian
Binghamton University
133 PUBLICATIONS **1,716** CITATIONS

SEE PROFILE

All content following this page was uploaded by Montri Karnjanadecha on 03 January 2014.

The user has requested enhancement of the downloaded file.

Signal Modeling for High-Performance Robust Isolated Word Recognition

Montri Karnjanadecha and Stephen A. Zahorian

Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529, USA

ABSTRACT

This paper describes speech signal modeling techniques which are well suited to high performance and robust isolated word recognition. We present new techniques for incorporating spectral/temporal information as a function of temporal position within each word. In particular, spectral/temporal parameters are computed using both variable length blocks with a variable spacing between blocks. We tested features computed with these methods using an alphabet recognition task based on the ISOLET database. The Hidden Markov Model Toolkit (HTK) was used to implement the isolated word recognizer with whole word HMM models. The best accuracy achieved for speaker independent alphabet recognition, using 50 features, was 97.9%, which represents a new benchmark for this task. We also tested these methods with deliberate signal degradation using additive Gaussian noise and telephone band limiting and found that the recognition degrades gracefully and to a smaller degree than for control cases based on MFCC coefficients and delta cepstra terms.

EDICS: 1-RECO
Correspondence author: Stephen A. Zahorian
Telephone: 757-683-3745
Fax: 757-683-3220
Email: szahoria@odu.edu

I. INTRODUCTION

Continuous speech recognition systems have been developed for many applications, since low-cost PCs now have sufficient computing power for speech recognition software. However, high performance and robust isolated word recognition, particularly for the letters of the alphabet and for digits, is still useful for many applications such as recognizing telephone numbers, spelled names and addresses, ZIP codes, and as a spelling mode for use with difficult words and out-of-vocabulary items in a continuous speech recognizer.

Because of the potential applications, as mentioned above, many isolated word recognizers are optimized for the digits or alphabet or both (alphadigit). The alphabet recognition task is particularly difficult because there are many highly confusable letters in the alphabet set - for example, the letters of the E-set, (/B, C, D, E, G, P, T, V, Z/), and the /M, N/ pair. Also, since language models generally cannot be used, the alphabet recognition task is a small, challenging, and potentially useful problem for evaluating acoustic signal modeling and word recognition methods.

The ISOLET database [2] was used for all experiments reported in this paper. This database was intended for evaluation of isolated word recognizers; therefore, it has been used in several studies. Thus, it is possible to directly compare results. All of the experiments in the present study[1] were performed in a speaker-independent fashion using all files from the database, i.e., 120 speakers (60 males and 60 females) for training and 30 speakers for testing (15 males and 15 females). The Hidden Markov Model Toolkit (HTK) [14], was used to implement the HMM recognizer. The best result obtained in our study of 97.9% corresponds to 19% fewer errors than the best result reported in the literature for this task from other labs [10], and 47% fewer errors than the next best reported result [1]. More importantly, the method introduced in this paper is easier to implement and duplicate than these previous state-of-the-art systems.

This paper is organized as follows. Section II briefly mentions some issues in speech signal modeling methods and gives some results from the literature for the alphabet recognition task reported

on in this paper. Section III describes the signal modeling methods used in this study. Experimental verification is presented in Section IV, and conclusions are drawn in Section V.

II. BACKGROUND

One of the most thorough tests of various front ends for use with automatic speech recognition is the often-quoted paper by Davis *et al.* [3]. In this paper, the signal representations Mel Frequency Cepstrum Coefficients (MFCC), Linear Frequency Cepstrum Coefficients (LFCC), Linear Prediction Coefficients (LPC), Linear Prediction Cepstrum Coefficients (LPCC), and Reflection Coefficients (RC) were all tested for effects on word recognition accuracy with a template-based, dynamic time warping recognizer. The primary conclusion of this paper was that the MFCC parameters perform the best overall for automatic speech recognition (ASR), with about six coefficients capturing most information relevant for speech recognition, but with some increases in ASR performance as up to 10 coefficients are used. Although the Davis *et al.* work would seem to be somewhat dated, as it is now almost 20 years old and there have been many refinements in speech signal representations, the basic conclusions of this paper are still considered valid.

In a more recent tutorial paper [13], Picone summarizes and compares some of the work on speech signal representations that has been done in the1980s and early 1990s. In a survey of 31 reported ASR systems, 21 used some form of cepstral coefficients as the basic signal features, with FFT-derived MFCC the most common type. He did note that many systems combine multiple frames of speech parameters, usually over intervals ranging from 30 to 75 ms, to compute additional parameters which capture spectral/temporal information that improves ASR. A signal parameter vector consisting of cepstrum coefficients, the first derivative of the cepstrum coefficients, power, and the derivative of the power was reported as the *de facto* standard for ASR. This “standard” is still widely used; for example, the front end of the HTK, which we use as a control in this paper, also uses this method.

Due to the importance of temporal information for HMM-based speech recognizers, Milner [11] investigated and generalized several methods for computer spectral/temporal features. Experimental results showed that many temporal encoding methods, including the DCT method, outperformed the static features augmented with delta and delta-delta terms. Best performance was achieved with the Karhunen Lueve transform. In other related work, Harte *et al.* [6] showed that a feature vector with components from both short intervals and longer time intervals (segments) resulted in superior performance for the highly confusable E-set versus features computed from a single time interval.

The best speaker-independent performance on OGI's ISOLET database with the same test data as used for our work (97.37%) was obtained using a 2-stage, phoneme-based, context-dependent HMM recognizer [10]. The features used were 26 MFCC terms consisting of 12 weighted MFCC terms, 12 delta MFCC terms, a normalized energy term and a delta energy term. The next best reported result of 96.0% was obtained using fully connected feed-forward neural networks with 617 inputs and 26 outputs [1]. Those 617 inputs were extracted from selected portions of tokens whose boundaries were previously determined by a rule-based segmentor. Many other studies of alphabet recognition are summarized in [10].

III. METHOD

A. Signal modeling

The first stages of the signal modeling method used in the present work are a variation of the method used in [15] and [16] for phonetic classification. We summarize the method and discuss in more detail some changes that make the method especially suited to high-accuracy isolated word recognition.

The basic method begins with second-order pre-emphasis with a second-order filter centered at 3200Hz. Next, Kaiser-windowed (beta = 6) 20-ms speech frames are analyzed with a 512-point FFT every 5 ms. The spectral range was limited to 60 dB for each frame, using a floor. Neither envelope tracking nor a morphological filter (in contrast to [15] and [16]) was used. From this "preprocessed"

spectra, a modified discrete cosine transform over frequency is computed to obtain a set of Discrete Cosine Transform Coefficients (DCTCs) which compactly represent the spectral shape of each speech frame. The basis vectors used in the DCTC calculations are modified so that the frequency resolution approximates a Mel frequency scale. In the final step, the DCTCs are block-encoded with a sliding overlapping block using another cosine transform over time that was used to compactly represent the trajectory of each DCTC. The cosine basis vectors in this second transform are also "modified" so that the temporal resolution was better near the middle portion of each block relative to the endpoints. The coefficients of this second transform are called Discrete Cosine Series Coefficients (DCSC). The overall processing is similar to calculation of cepstral coefficients followed by "delta" term calculations. However, the "new" method is very flexible with a small number of parameters which control the details of the analysis, particularly in terms of spectral/temporal frequency resolution tradeoffs.

The method outlined above was used to compute 10 DCTC terms using a bilinear warping factor of .45 over the frequency range of 60 Hz to 7600 Hz (or 300 Hz to 3200 Hz for simulated telephone speech). These 10 terms were in turn each represented by a 5-term modified cosine expansion over time, resulting in a total of 50 parameters (DCSCs) to encode each block. In the next few paragraphs, we describe the methods used to determine the time duration on which each of the segment features is based, and which we call the *block length*, or number of frames used to compute each set of 50 features, and the time spacing between blocks.

Using this formulation, it is quite straightforward to manipulate the block length and/or block spacing based on the signal properties. Shorter block lengths and block spacings could be used to achieve better temporal resolution, and longer block lengths and spacings could be used to increase temporal smoothing and reduce redundancy. Thus, the block length presumably should be short in regions where the spectrum is rapidly changing, such as the initial portions of most of the words in the ISOLET database, and much longer in the vowel portions of each word.

The main objective of the current paper is to report two methods for adapting the block length based on the signal properties, as applied to isolated words. The two methods investigated were:

1) Variable block length method

For this method, the block length in the DCSC calculations was varied according to position within the utterance. In particular, at the beginning of an analyzed token, a block size of 45 ms was used. As the analysis window moved forward, the block size increased until a maximum of 215 ms was reached. The block size was then fixed at this maximum until the end region of the utterance was reached. At this point the block length was again gradually reduced until, for the very final block, it again reached 45 ms. Time "warping" was also applied to each block, with the amount of warping controlled by the beta value for a Kaiser window, and with this beta value linearly interpolated from 0 for the shortest length windows, up to a maximum value of 5 (approximately a Hanning window) for the longest blocks.[2] Thus, the features gave better time resolution for the onset and offset portions of each word and less time resolution in the central portions of each word. The block features were re-computed every 10 ms. No manual segmentation or phonetic labeling was required or used. The primary modification, relative to [8] and [9], is that the block length was varied at both ends of each analyzed utterance, rather than only for the beginning section. See Fig.1 for an illustration of this variable block length method.

Each gray horizontal bar in Fig. 1 represents a block. The alignment over the frequency axis of the blocks has no relation to frequency---it is used only to illustrate the progression of the blocks. Note that although the main effect is the variation in block length at each end of the utterance, the block spacing is also effectively twice as long in the center of the utterance, relative to the end regions. A drawback of this method is that it is not obviously extendable to continuous speech recognition.

2) Variable block spacing method

This method adjusts block spacing to accommodate the spectral characteristics while the block size is constant. The objective is to advance the block by a small amount in the regions where the spectrum is changing rapidly and to advance the block by a large amount when the spectrum is changing slowly. This yields a non-uniform data rate. In this work the spectral rate of change was quantified using the spectral derivative method introduced by Furui [5].

The overall method is implemented as follows. First, frame-based features of the utterance are computed and the spectral derivative is determined. Then block spacing is calculated to be proportional to the spectral derivative, with a proportionality constant set to obtain a specified maximum and minimum block length. Note that the spectral derivative was clipped at an empirically determined threshold (constant threshold over all utterances), which was determined such that only about 5% of spectral derivative calculations resulted in clipping. The DCT over time was then applied to each block of fixed length, but using block centers as determined from the block spacing step.

Figure 2 depicts an example of this method when applied to a real speech utterance. Note that the block spacing is relatively short at the beginning and end of the utterance and long in the central portion.

B. Endpoint detection

For all experiments, an endpoint detection adapted from the one given in [4] was used to locate endpoints, which were then extended 30 ms in each direction (i.e., backward in time for the onset and forward in time for the offset) to allow for some inaccuracies in the automatic detection, and also to include a small amount of silence at the beginning and end of each utterance. The primary difference between the method given in [4] and our implementation was that we used 20 ms frames for the first pass of endpoint detection, and 10 ms frames for the second pass, as opposed to the longer frames used in the original method. This endpoint detection method resulted in approximately a 34% decrease in

errors for an analysis condition corresponding to the best case reported here. However, use of the endpoint algorithm, but without the 30 ms silence added at each end, resulted in an error rate more than doubled for the same analysis conditions.

C. Linear Discriminant Analysis

In some of the experiments reported in this paper, LDA was used to transform and reduce the size of speech feature sets. In our implementation of LDA, we compute two covariance matrices, the between-class covariance matrix B and the within-class covariance matrix W. The B matrix is estimated as the grand covariance matrix of all the training data (the same as for a principal components analysis). The W matrix is estimated by computing the average covariance of time-aligned frames of data belonging to the same class. Time alignment is accomplished using dynamic time warping to determine a "target" for each word by successively aligning and averaging all tokens of that word in pairs until only one token remains. Covariance contributions are then computed as variations about the target, after another time alignment to that target. These two matrices are then used to create a linear discriminant analysis transformation which maximizes the ratio of between-to-within class covariance. Our implementation of this technique is similar to that presented in [7] and [12].

D. Hidden Markov Models

For all experiments presented in this paper, the HTK toolkit was used to implement a word-based HMM recognizer. In each experiment, there were 26 HMM models trained to recognize all 26 letters of the English alphabet. Except where otherwise mentioned, each model had 5 states and 3 multivariate Gaussian mixtures with a full covariance matrix. Only self transitions and transitions to the next state were allowed. In the training phase of each experiment, every training utterance was segmented into equal lengths and then initial model parameters were estimated. Next, the Viterbi decoding algorithm was applied to determine an optimum state sequence of each training token. Every token was re-segmented based on its corresponding optimum state sequence. Model parameters were re-

estimated repeatedly until the estimates were unchanged or the maximum number of iterations was reached. Note that no Baum-Welch iterations were performed, as they were not found to improve accuracy on test data. Again, the Viterbi algorithm was applied in the testing phase to determine the model that best matched each test utterance.

IV. EXPERIMENTAL VERIFICATION

A. Database

The ISOLET database from OGI [2] was used in all experiments. The database is comprised of the English alphabet letters spoken by 150 speakers, 75 males and 75 females. Each speaker uttered the same word twice. Thus there are a total of 7800 utterances. The database is divided into 5 groups of equal size: ISOLET-1, ISOLET-2, ISOLET-3, ISOLET-4, and ISOLET-5. Utterances were recorded as isolated words with a sampling frequency of 16000 Hz and a 16-bit A-to-D system. The speech signal-to-noise ratio (SNR) reported by OGI is 31.5 dB with a standard deviation of 5.6 dB. The endpoints were refined with the algorithm mentioned.

B. Experiments

Several speaker–independent recognition experiments were conducted with the alphabet set to evaluate the algorithms described above, and also to determine the effects of variations in some of the parameter values. Except where noted differently, all experiments used ISOLET-1 through ISOLET-4 for training and ISOLET-5 for testing. As a control, experiments were also conducted using the MFCC front end supplied with the HTK, using primarily default parameter values (pre-emphasis filter transfer function $H(z) = 1\text{-}0.95z^{-1}$, frame size of 25 ms, frame spacing of 10 ms, Hamming window, 24 filters, 39 total terms, including delta terms and delta-delta terms).[3] A brief description of these experiments and their results follows.

1) Baseline experiments: The first baseline experiment was based on single frame features for both DCTCs and MFCCs. Typical test results in terms of percent accuracy were 82.3% for 10 DCTC terms and 82.6% for 12 MFCCs + Energy (13 terms). In general, for all conditions tested (various frame lengths, numbers of terms, etc.) results for these two static feature sets were quite similar. The next baseline experiment was to use a fixed block length for both DCSC features and MFCC features. Typical results were 94.6% for the DCSC features and 95.8% for the MFCC features. These results were based on 50 terms for the DCSC features (10 DCTCs, each represented with 5 terms over a block length of 115 ms,) and 39 MFCC terms, with delta and delta-delta parameters computed using 5 frames (65 ms). Although several other parameter settings were tried (block length, total number of terms, etc.), the conditions mentioned were the best (by a small amount) of the ones tried for both the DCSC terms and the MFCC terms.

2) Variable block length experiment: As mentioned previously, the endpoint detection program included 30 ms of silence at both ends of each utterance in the database. However, as noted in section IV.B.3 (below), the best recognition accuracy on test data (ISOLET-5) of 97.9% was achieved using the 30 ms of silence before the detected onset of speech, and 25 ms of silence after the detected final endpoint. Therefore, unless otherwise noted, this configuration (30 ms initial silence, 25 ms final silence) was used in this experiment and all the following experiments.

In this test, the method introduced in Section III.A was evaluated using 10 DCTCs, each represented with 5 terms in the DCSC expansion (50 features), with the block length varied from 6 frames (45 ms) at the beginning and end of each utterance up to 40 frames (215 ms) at the center of each utterance. As verification, this test was repeated in "round robin" fashion, using ISOLET-1 through ISOLET-5 as test data one at a time (and the remaining four sets for training in each case). Results of the round robin test were then averaged. Similar testing was done with the 39 MFCC parameters mentioned above. Results for the four cases are given in Table I. Note that the DCSC and results obtained from the robin tests are very close to those obtained with ISOTLET-5 tests, thus implying that the parameters are not overly "tuned" to ISOLET-5.

3) Variable block spacing experiment: This method was tested with the same task, but using a different HMM configuration. Each letter was modeled with a 6-state, 3-diagonal mixture component and trained with the Viterbi and B&W algorithms. The block length used was 93 ms and the block spacing was varied from 2 to 16 ms. Each block was represented by 39 terms (13 DCTCs expanded with 3 DCSs). The highest accuracy obtained on test data (ISOLET-5) was 97.8%. Note this result was nearly as high as the best results of 97.9% given in Table 1 for the variable block length method. When the experiment was repeated using the full covariance matrix method, and the same HMM configuration as for experiment 2, results obtained were 96.6%. Despite the promise of this method using the relatively simple diagonal covariance matrix, it was not tested further since even better results were obtained with the variable block length method.

4) Endpoint examination experiment: Since the block length in the variable block length method depends on the position of the block relative to the determined starting and final endpoints of each word, we hypothesized that performance might depend heavily on the accuracy of the endpoint algorithm. This notion was reinforced when we repeated the test mentioned above with the variable block length, but using the data as distributed without the benefit of the refined endpoint algorithm, and found that accuracy dropped from 97.9% to 96.8% (about a 52% increase in errors.) To test our hypothesis more systematically, we independently varied the starting and final endpoints over a range of +-30 ms from the automatically computed location, with the other endpoint fixed (30 ms for starting endpoint and 25 ms for final endpoint). A negative amount of silence means that a portion of speech is discarded. With the exception of these variations in endpoints, all other signal processing was identical to that used above in the variable block length experiment. The recognition results for these tests are depicted in Figure 3.

The results clearly illustrate that, at least for this data, the beginning endpoint is much more critical than the final endpoint. It also appears that adding even more than 30 ms of silence before the onset of each word would have been beneficial; however, this could not be done since the original

tokens did not include sufficient extra silence. The absolute best result of 97.9% was obtained with 30 ms of initial silence and 25 ms of ending silence. These values of silence were thus used in the other experiments reported in this paper.

5) Signal to noise ratio experiment with and without LDA and with and without band limiting:

To test the robustness of the signal features more thoroughly, experiments were conducted with additive Gaussian white noise over a range of SNRs from –10dB to + 30 dB**.** For all experiments, noise was added to both training and testing data. These tests were done both with the 50 DCSC features, as used above, and the 39 MFCC features (also previously mentioned). All conditions were identical to those reported for Experiment 2, with the exception of the additive noise. Moreover, tests were made using LDA for each noise level, with the hypothesis that the effects of LDA might depend on noise level. Based on pilot experiments, we extracted 30 LDA terms for the DCSC case and 25 LDA terms for the MFCC case (about 60% of the size of the original feature vector in each case). Results are given in Fig. 4.

Note that for all cases except SNRs of –10 and –5 dB, the 50 DCSCs perform better than the 39 MFCC parameters. For both features sets, LDA is beneficial with SNRs of 0 dB or worse. For higher SNR values, the effect of LDA is quite small for both feature sets, sometimes slightly degrading performance and sometimes slightly improving performance. For "clean" speech, SNRs of 25 dB or higher, LDA degrades the DCSC features - but only by a small amount - whereas LDA improves the MFCC features nearly to the level of DCSC features.

As one final "robustness" test, tests were made with band-limited speech (300 Hz to 3200 Hz) to simulate telephone bandwidth for each noise level. Results are shown in Fig. 5.

In every case, without the use of LDA, the DCSCs resulted in higher accuracy than the MFCCs. Except at an SNR of 30 dB for the DCSCs, the performances of DCSCs and MFCCs were improved somewhat with LDA. In general the performance increase due to LDA was higher at lower values of SNR.

The effects of LDA applied to features extracted from band-limited speech are quite different depending on DCSC versus MFCC parameters. For almost all noise levels, the LDA results in very little change for the DCSC case, whereas the LDA results in larger improvements with the MFCC parameters. The performance obtained with LDA transformed MFCCs very closely matches that obtained with the DCSCs, with or without LDA.

Note that for the full bandwidth speech, the rates based on DCSC signal modeling are typically about 1.5% higher than the rates based on MFCC signal modeling. For the telephone bandwidth case, the DCSC method averages about 3.4% higher than for the MFCC method. Additionally, the typical degradation between full bandwidth and telephone bandwidth is also less for the DCSC case versus the MFCC case (average degradation of 5.3% versus 7.3%).

C. Error analysis

Table 2 shows a subset of the confusion matrix for the highest accuracy case (97.9% accuracy on clean, full bandwidth test data with 50 DCSCs). In the matrix, each row represents the "actual" spoken letter, and each column represents the identified letter. This confusion matrix is given only for the E-set letters and the /M,N/ pair, since these 11 letters accounted for about 2/3 of the total errors (23 errors out of a total of 33). The single most confusable pair was the /M,N/ pair.

The number of errors was small enough to be inspected visually, as plots of time waveforms and spectrograms on a computer, and by listening. Of these 33 errors, 15 were due to confusions within the E set, eight were due to confusions between letter M and N, and only ten confusions were among the remaining 14 letters. After listening to all of the error tokens, we concluded there are four situations for which tokens were misrecognized. Six tokens appeared to have a severe endpoint detection problem. There were nine tokens pronounced in an unusual way, mostly by a single speaker. Also, there were four tokens misrecognized because they are so similar to other letters that, even with careful listening, they were difficult to recognize. Finally, fourteen tokens sounded intelligible and distinct with no

obvious reasons for the errors in machine performance, although some of these may have had inaccuracies in endpoint detection.

To test the hypothesis that several of the errors were due to endpoint problems, we manually corrected the endpoints for all the error tokens. A recognition test performed on this corrected test data results in 19 fewer errors, or 99.1% correct. Although this result does not really "count," since it was not done fully automatically, it does help illustrate the importance of good endpoint detection.

V. CONCLUSIONS

The variable block length signal analysis method with 50 DCSC terms and a "standard" (but full-covariance) HMM recognizer results in 97.9% accuracy for the alphabet set. This represents a 19% reduction in errors relative to the best previously reported result in the literature [10] for the same database. The method used in the current work would be quite easy to duplicate—the front end signal processing consists of dot product operations between blocks of FFT computed log spectrum with cosine-like basis vectors over time and frequency, with simple procedures for adjusting the block length and spacing, and a one stage whole word HMM. In contrast, the two methods with the best previously reported results used either much more complex features (617 features in [1], or a 2-stage context-dependent phonetic HMM [10].) The best result obtained with the variable block spacing method (97.8%) was slightly lower, but still better than the best previously reported result, and it was obtained using the computationally advantageous diagonal-covariance matrix HMM. The general signal modeling approach used in this work, block-encoded DCTCs based on a cosine transform over time, is generally more robust to noise and band limiting than are MFCC terms augmented by delta and delta-delta MFCC.

The primary contribution of this paper is the demonstration that the calculation of signal trajectories over intervals with lengths or spacings dependent on position within the utterance (either based on signal properties such as spectral derivative, or "anchor" points such as the endpoints) can

improve performance. The advantage of this method is that it is able to capture rapid transients at the beginning and ending of each word, while simultaneously using longer, more noise resistant averaging intervals in the center of each word. We also showed that the benefit of using LDA for automatic speech recognition depends heavily on the features and noise level. The improvements in ASR performance due to LDA tend to be the largest under noisy conditions, and with signal parameters that are sub optimal in terms of recognizer performance. The only real benefit of LDA for "good" parameters, such as the DCSCs used in this study, is that a reduction in dimensionality by about a factor of two is possible with very little change in performance. The results in this paper also demonstrate that accurate endpoint detection is still an area for further improvement for very high performance isolated word recognition systems.

The variable block spacing method, with block spacing dependent on spectral derivative, would appear to be more easily extensible to continuous speech recognition. An investigation of this method with continuous speech recognition remains a topic for further investigation.

ACKNOWLEDGMENT

Portions of this work were supported by NSF grant BES-9977260.

Footnotes

1. A preliminary version of this work was reported in [8] and [9].

2. The values for minimum and maximum block length and degree of time warping were varied and tested. The values mentioned, obtained from pilot experiments, were the ones used for our experimental results.

3. Note that one difference between this control signal processing and the "new" methods presented in this paper is the pre-emphasis filter. The first-order filter was used for the control, since this is typically used. In our own previous work ([15], [16]), we have reported that the second-order pre-

emphasis is marginally better than the first order pre-emphasis (and better matches psychophysical data), and thus have incorporated this filter as a “standard” for our signal processing.)

REFERENCES

[1] Cole, R., Fanty, M., and Muthusamy, Y., "Speaker-independent recognition of spoken English letters," *Proc. Int. Joint Conf. Neural Networks*, Vol.2, pp. 45-51, June 1990.

[2] Cole, R., Muthusamy, Y., and Fanty, M., "The ISOLET spoken letter database," *Tech. Rep. 90-004*, Oregon Graduate Inst., 1990.

[3] Davis S. B., and Mermelstein, P., "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-28, pp. 357-366, 1980.

[4] Dermatas, E. S., Fakotakis, N. D., and Kokkinakis, G. K., "Fast Endpoint Detection Algorithm for Isolated Word Recognition in an Office Environment," *Proc. ICASSP 91*, Toronto, Canada, pp. 733-736, 1991.

[5] Furui, S., "On the role of spectral transition for speech perception," *J. Acoust. Soc. Am*. 80, pp. 1016-1025, October 1986.

[6] Harte, N., Vaseghi, S., Milner, B., "Dynamic Features for Segmental Speech Recognition," Proc. ICSLP 96, Part 2, Vol. 2, pp. 933-936, 1996.

[7] Hunt, M. J., and Lefebvre, C., "A comparison of several acoustic representations for speech recognition with degraded and undegraded speech," *Proc. ICASSP 89*, Glasgow, Scotland, pp. 262- 265, May 1989.

[8] Karnjanadecha, M., and Zahorian, S. A., "Robust Feature Extraction for Alphabet Recognition," *Proc. ICSLP* 98, Sydney, Australia, Vol. 2, pp. 337-340, 1998.

[9] Karnjanadecha, M., and Zahorian, S. A., "Signal Modeling for Isolated Word Recognition," *Proc. ICASSP* 99, Phoenix, AZ., Vol. 1, pp. 293-296, March 1999.

[10] Loizou, P. C., and Spanias, A. S., "High-Performance Alphabet Recognition," *IEEE Trans. Speech and Audio Processing*. Vol. 4, pp. 430-445, Aug. 1996.

[11] Milner, B., “Inclusion of Temporal Information into Features for Speech Recognition,” *Proc. ICSLP 96*, pp. 256-259, 1996.

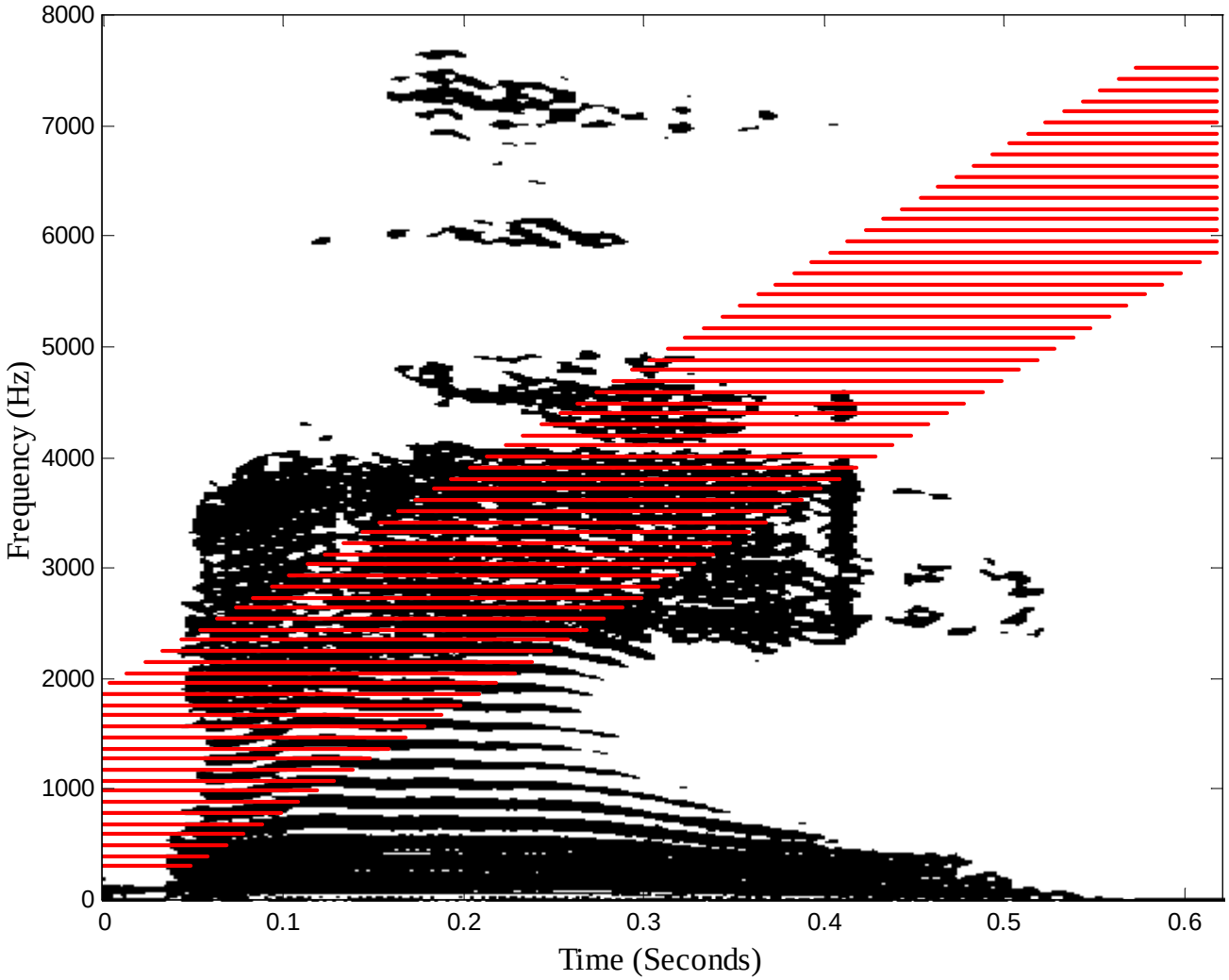
[12] Parsons, T., *Voice and Speech Processing*, Mc-Graw Hill, New York, 1987.

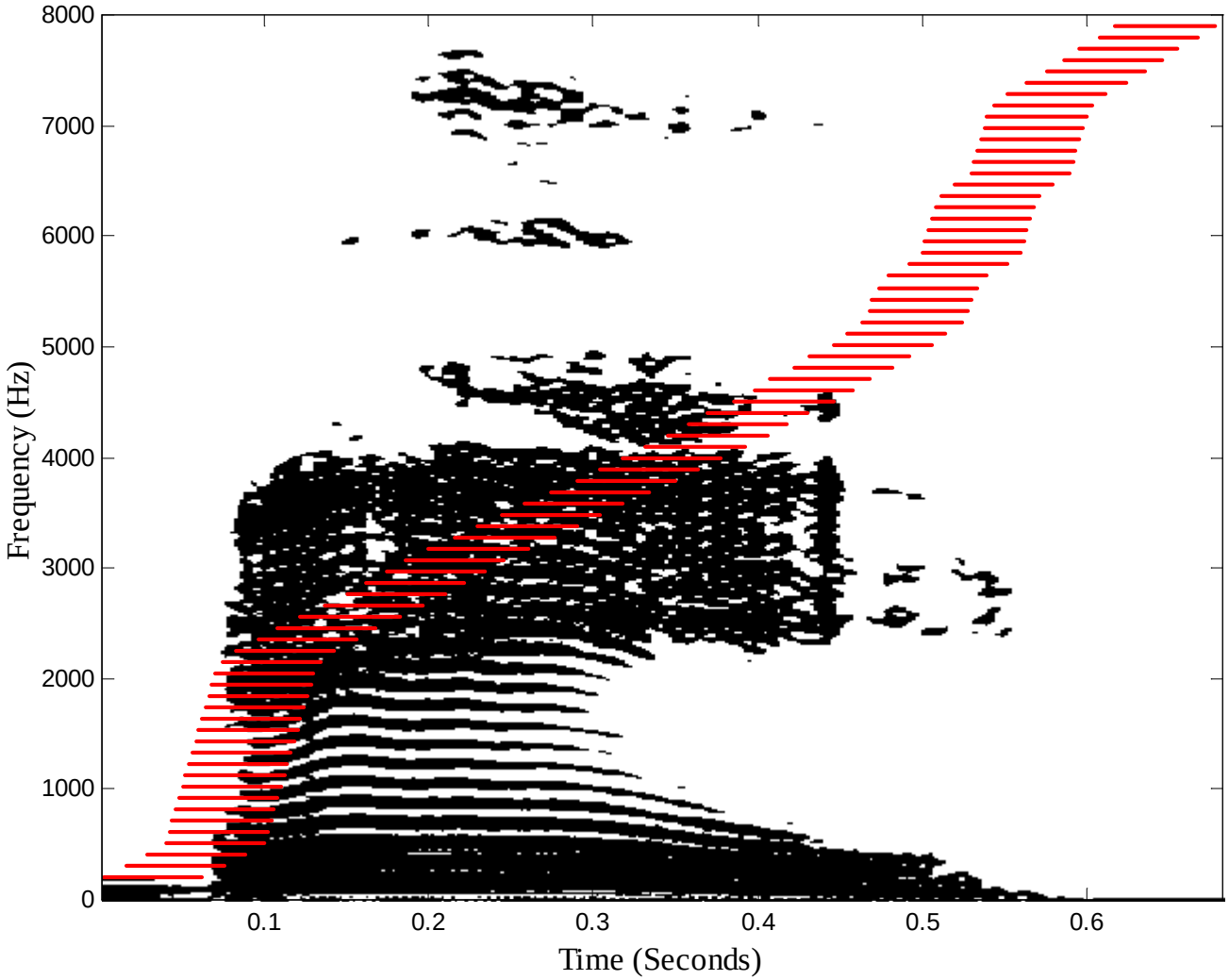
[13] Picone J., “Signal Modeling Techniques in Speech Recognition,” *IEEE Proceedings*, vol. 81, pp. 1215-1247, Sept. 1993.

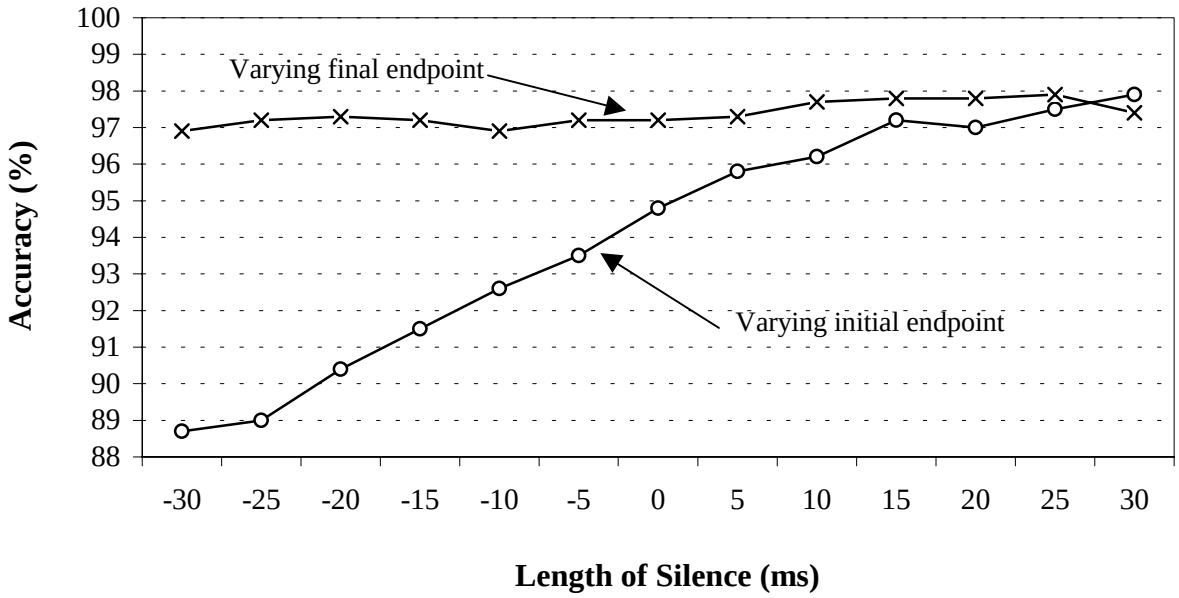
[14] Young, S. J., Odell, J., Ollason, D., Valtchev, V., and Woodland, P., “Hidden Markov Model Toolkit V2.1 reference manual,” *Technical report*, Speech group, Cambridge University Engineering Department, March 1997.

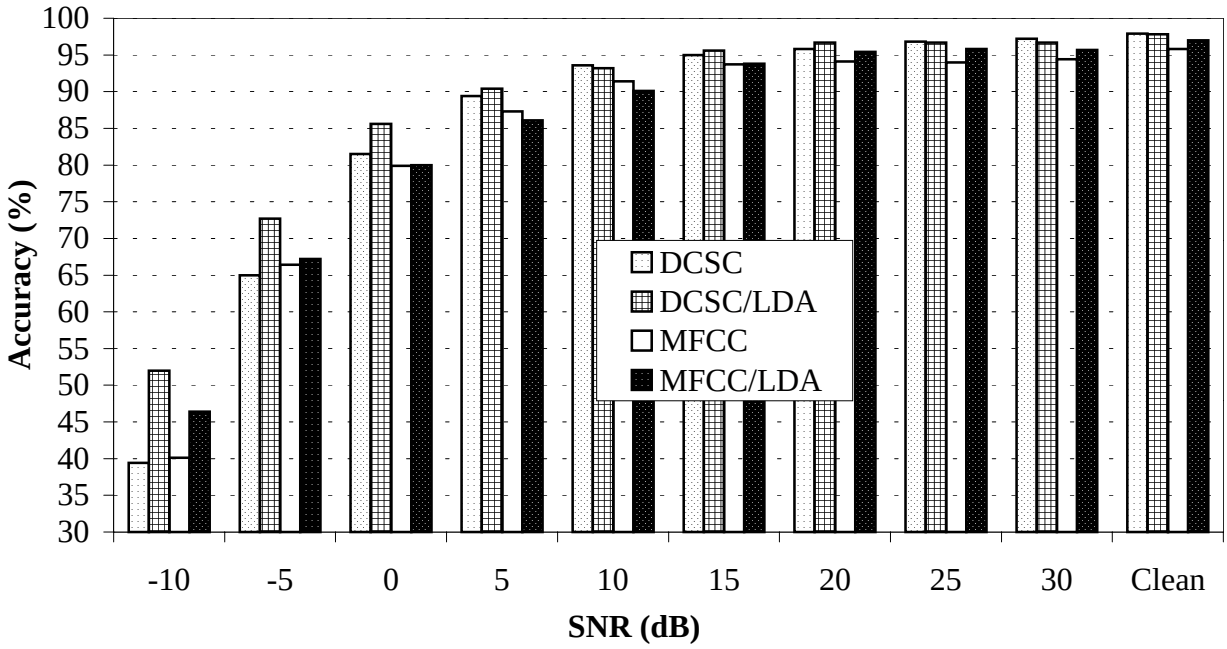
[15] Zahorian S. A., and Nossair Z. B., “A Partitioned Neural Network Approach for Vowel Classification Using Smoothed Time/Frequency Features*,” IEEE Trans. Speech and Audio Processing*, Vol. 7, pp. 414-425, July 1999.

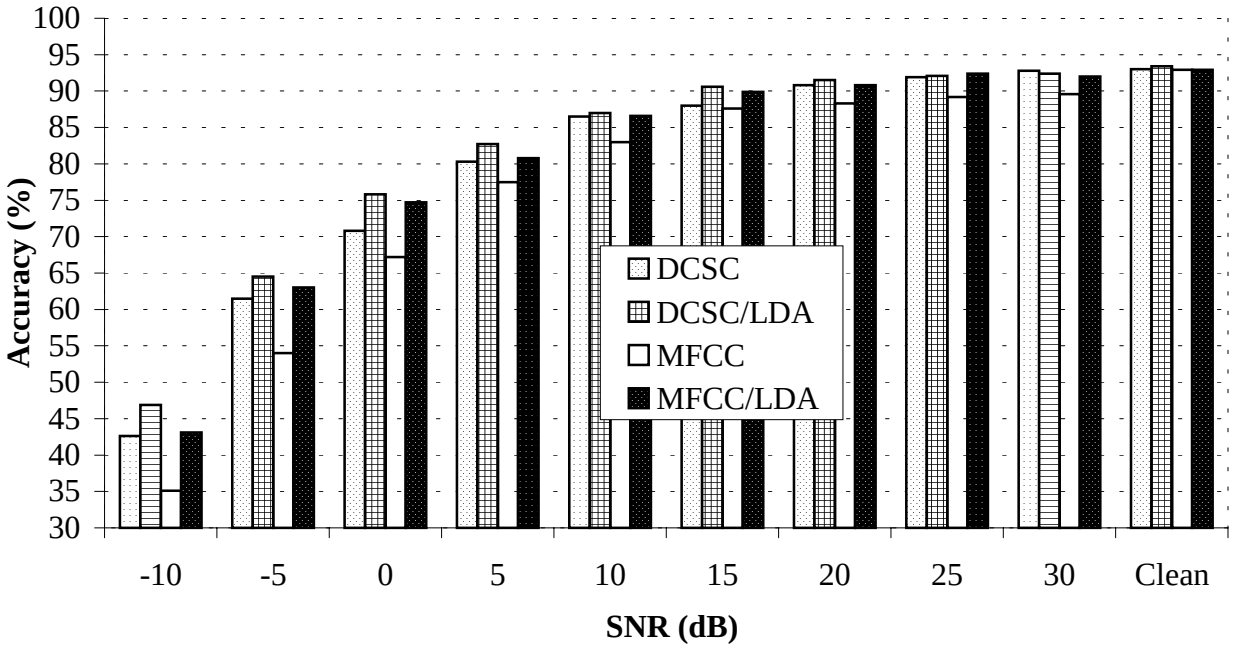
[16] Zahorian, S. A., Silsbee, P. L., and Wang, X., "Phone Classification with Segmental Features and a Binary-Pair Partitioned Neural Network Classifier," *Proc. ICASSP 97*, Munich, Germany, pp. 1011-1014, April.1997.


8000
7000
6000
5000
4000
3000
2000
1000
0
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
Time (Seconds)


8000
7000
6000
5000
4000
3000
2000
1000
0
Frequency (Hz)
0.1
0.2
0.3
0.4
0.5
0.6
Time (Seconds)


100
99
98
97
96
95
94
93
92
91
90
89
88
Accuracy (%)
Varying final endpoint
Varying initial endpoint
-30
-25
-20
-15
-10
-5
0
5
10
15
20
25
30
Length of Silence (ms)


100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
Accuracy (%)
DCSC
DCSC/LDA
MFCC
MFCC/LDA
-10
-5
0
5
10
15
20
25
30
Clean
SNR (dB)


100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
Accuracy (%)
DCSC
DCSC/LDA
MFCC
MFCC/LDA
-10
-5
0
5
10
15
20
25
30
Clean
SNR (dB)

Type of Features	Recognition Accuracy (%)	
	ISOLET-5 for Test	Round Robin Test
DCSC	97.9	97.7
MFCC	95.8	95.9

	B	C	D	E	G	M	N	P	T	V	Z
B	59									1	
C		59									1
D			58		1				1		
E	1			59							
G					56				2		
M						58	2				
N						6	53				
P			1					59			
T					2				56		
V										59	1
Z		3								1	56

View publication stats