SPEECH/SILENCE SEGMENTATION FOR REAL-TIME CODING VIA RULE BASED ADAPTIVE ENDPOINT DETECTION

J. F. Lynch Jr.
J. G. Josenhans
R. E. Crochiere

Speech Processing Department

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT: A new algorithmic technique is presented for efficiently implementing the end-point decisions necessary to separate and segment speech from noisy background environments. The algorithm utilizes a set of computationally efficient production rules that are used to generate speech and noise metrics continuously from the input speech waveform. These production rules are based on statistical assumptions about the characteristics of the speech and noise waveform and are generated via time-domain processing to achieve a zero delay decision. An end-pointer compares the speech and silence metrics using an adaptive thresholding scheme with a hysteresis characteristic to control the switching speed of the speech/silence decision.

The paper further describes the application of this algorithm to silence compression of speech in which speech segments are encoded with a low bit-rate encoding scheme and silence information is characterized by a set of parameters. In the receiver the resulting packetized speech is reconstructed by decoding the speech segments and reconstructing the silence intervals through a noise substitution process in which the amplitude and duration of background noise is defined by the silence parameters. A noise generation technique is described which utilizes an 18th order polynomial to generate a spectrally flat pseudo-random sequence that is filtered to match the mean coloration of acoustical background noise. A technique is further described in which the speech/silence transitions are merged rather than switched to achieve maximum subjective performance of the compression technique.

The above silence compression algorithm has been implemented in a single DSP-20 signal processing chip using sub-band coding for speech encoding. Using this system, experiments were conducted to evaluate the performance of the technique and to verify the robustness of the endpoint and silence compression over a wide range of background noise conditions.

I. Introduction

This paper presents a technique for separating audio signals into speech and silence segments and efficiently encoding the silent segments. When added to a popular speech coder [5-7] this "silence compression" technique can significantly reduce the average bandwidth required to record speech. The silence compression technique operates in real time and requires no additional delay to be inserted in the transmission path.

The solution presented here uses a heuristic set of computationally efficient production rules to continuously generate speech and noise metrics from the input signal. The metrics are continuously updated each sample period and then compared to each other to generate a speech/silence decision with no delay from the input. During silent segments the level and duration of the noise is encoded and later used by the decoder to reconstruct these intervals with simulated noise.

II. Speech and Noise Metric Generation

Silence compression uses heuristic production rules to generate speech and noise metrics continuously from the input signal. These metrics are used later to generate the causal speech/noise decision. The first step to designing these production rules is to state some known properties about the speech and noise signals to provide a foundation for the design.

The following statements give a few properties of the speech signal that will be put to use later in the production rules:

1. Empirical evidence shows that 99.9% of "continuous speech" segments contain "talk-spurts" of less than 2.0 seconds in duration [1].
2. Empirical evidence also shows that 99.56% of "continuous speech" segments have gaps of less than 150 ms. in duration [1].
3. Speech energy can only increase the signal level above the background acoustic level.

Statements about the expected nature of the "background acoustic noise" are more difficult to make since it is implicitly undefined. However in a practical sense the definition can be narrowed. Here, background noise is assumed to be the relatively stationary "hum" or "buzz" due to ventilating systems, computer equipment or the aggregate background sounds of an office environment. Nonstationary background sounds such as nearby conversations among workers and nearby ringing telephones are classed as speech sounds and not background noise. The following statements can be made about background noise in this context:

1. Background acoustic noise energy decreases with frequency at approximately 5 dB per octave [2].
2. A 60 Hz high pass filter is used in the system to remove sub-sonic background noise.
3. The background acoustic noise energy is relatively stationary if measured as consecutive energy sample windows of a few milliseconds in width.
4. Background acoustic noise is not generally stationary when comparing energy levels displaced by several seconds.

Figure 1 shows a picture of the average long-term power density spectrum for speech with the -5 dB per octave "Hoth" noise templates overlaid. Note that at frequencies below 400 Hz the speech energy is decreasing and the noise energy is increasing. Therefore it is possible to increase discrimination between the speech and noise signals by preemphasizing the input with a high pass filter. The following filter serves this function:

$$v(k) = i(k) - .95i(k-1) \qquad (1)$$

In this equation i(k) is the unprocessed input to the system and v(k) is the preemphasized output.

1348 CH2396-0/87/0000-1348 $1.00

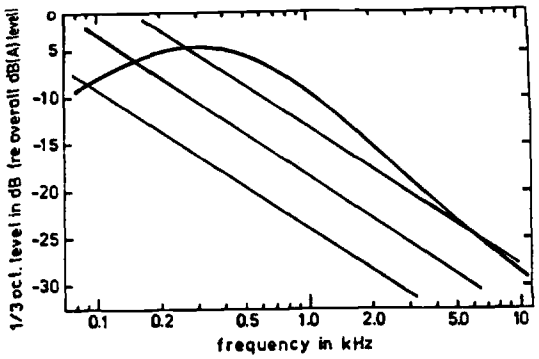

Fig. 1 Long Term Speech Spectrum showing Hoth Noise Curves

The purpose of generating a speech metric is to provide a scalar quantity which exhibits the current energy level of the speech so that it may be compared to the noise metric to determine the beginning and ending point of continuous speech segments. As stated previously, continuous speech contains short (less than 150 ms.) intersyllabic gaps and therefore the speech metric must "bridge over" these gaps to prevent switching during continuous speech. To simply perform these functions the following production rules are used:

$$u(k) = |v(k)| \tag{2}$$

$$\text{if}(s(k) > u(k)) \quad s(k) = u(k) \tag{3}$$

$$\text{if}(s(k) \leqslant u(k)) \quad s(k) = (1-B_s)u(k) + B_s s(k-1) \tag{4}$$

In equation 2 the absolute value function is used instead of an energy calculation ($v(k)v(k)$) to generate the positive value that is used in the following production rules. A true energy metric would be more mathematically tractable but suffers from dynamic range limitations when generated in fixed point machines because of the squaring operation. The absolute value operation will give the same results so long as all other metrics are consistent. Equation 3 generates the exponentially mapped past indication of the peaks of the input waveform. The constant B_s in equation 3 is the decay time constant and is set to .9992 yielding a decay rate of 150 ms. at an 8000 Hz sampling rate. This decay time constant is sufficient to bridge the intersyllabic gaps present in continuous speech as shown in figure 2.

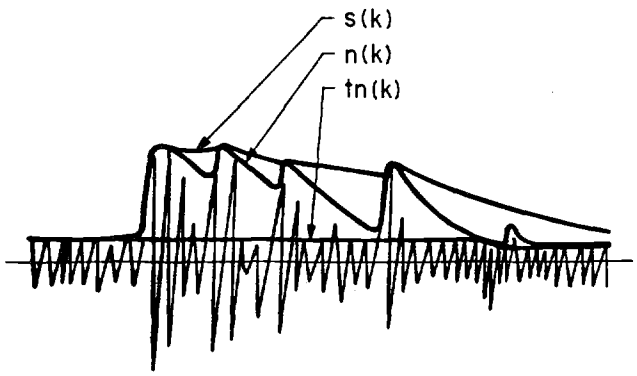

Fig. 2 Input Waveform Showing Speech and Noise Metrics

The production rules for the noise metric generate a single scalar quantity that indicates the current level of the background noise even during speech segments. This continuous adaptation property is important if the noise metric is to be able to adapt to any new noise environment. The first step in designing the noise metric is to consider the case when only noise is present. In this case, the noise metric should indicate the peak level of the noise to be consistent with the speech metric given above.

$$\text{if}(n(k) > u(k)) \quad n(k) = u(k) \tag{5}$$

$$\text{if}(n(k) \leqslant u(k)) \quad n(k) = (1-B_n)u(k) + B_n n(k) \tag{6}$$

These production rules generate the variable n(k) which indicates the peak noise level in the absence of speech. As can be seen in figure 2 n(k) is the exponentially mapped past value of the peaks of u(k). In these equations B_n is set equal to .9922 yielding a 16 ms. decay rate which is sufficient to bridge the widest gaps between noise peaks that can occur above the 60 Hz cutoff frequency.

In the presence of speech n(k) is inaccurate and tracks the speech levels returning to the noise level quickly during speech gaps. Therefore, to create a true noise metric, a method is necessary to allow adaptation to continue during speech but not be affected by the speech. Fortunately it is extremely rare for a talk-spurt (continuous burst of speech energy) to last longer than 2 seconds before a gap occurs even during "continuously spoken speech" [1]. A self evident theory given above states that speech energy can only cause a transitory increase in the input waveform level. Therefore if the noise metric is prevented from increasing until n(k) has increased for at least 2 seconds the noise metric will not be affected by speech. These production rules are implemented as follows:

$$\text{if}(tn(k) > n(k)) \quad tn(k) = (1-B_t)n(k) + B_t tn(k) \tag{7}$$

$$\text{if}(tn(k) \leqslant n(k)) \quad tn(k) = n(k) \tag{8}$$

Equations 7 and 8 generate the exponentially mapped past minimum of the noise level where the time constant B_t is set to .999975 yielding a time constant of 5 seconds. This long time constant was chosen because it is sufficient to bridge under the longest expected talk-spurt of 2 seconds, although some error is introduced into the noise metric tn(k) during the longest talk-spurts. The only advantage for using an exponentially mapped past to finding the minimum value of n(k) is its ease of implementation in a signal processor as a filtering operation.

Equation 8 adds an important feature which allows the noise metric tn(k) to decrease instantly with n(k) therefore allowing the noise metric to be updated with the regularity of the gaps between talk-spurts. This is possible since n(k) is always greater than or equal to the actual noise level, therefore tn(k) should always be less than or equal to n(k).

III. Speech/Silence Decision

Using the speech metric s(k) and the silence metric tn(k) generated above it is now an easy task to determine if the current input is speech or silence. The desired response of these heuristic rules is to make accurate speech/silence segmentation decisions with no delay, and to provide stable and jitter free segmentation over a wide range of conditions. The following rules implement the speech/silence decision:

$$\text{if}(s(k) > T_s tn(k) + T_{min}) \quad \text{declare speech segment} \tag{9}$$

$$\text{if}(s(k) < T_n tn(k) + T_{min}) \quad \text{declare noise segment} \tag{10}$$

$$\text{if}(T_n tn(k) + T_{min} \leqslant s(k) \leqslant T_s tn(k) + T_{min}) \quad \text{no change} \tag{11}$$

Figure 3 shows the graph of this switching characteristic that results using the above rules. T_s and T_n are the speech and noise thresholds and are set equal to 2.0 and 1.414 respectively. T_{min} dominates the right side of equations 10 and 11 when tn(k) is small and is set to a level that is 40 dB below the peak of the largest allowable input signal. Inputs smaller than T_{min} will always be segmented as noise. Hysteresis is described in equation 11 above and shown in figure 3 as the "no change" band. Hysteresis prevents the occurrence of switching transients at the edges of speech or silence segments when noise would otherwise cause rapid transitions between the speech and silence states.

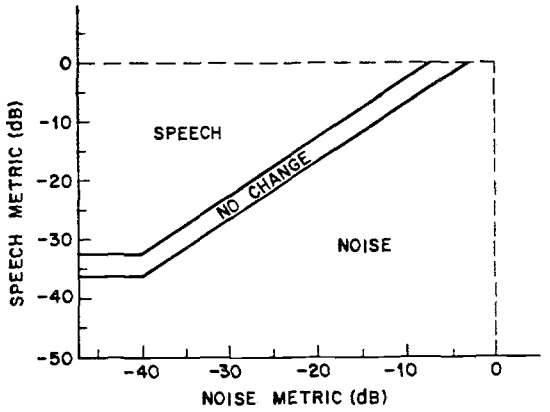

Fig. 3 Switching Characteristic using the Speech and Noise Metrics

IV. Silence Encoding and Reconstruction

The foregoing discussion has described the techniques used to segment the input waveform into its speech and noise segments. The use of this is to identify and encode the silence segments more efficiently with a "silence coder" than would otherwise have been done by the speech coder. Silence encoding saves the duration and noise level (tn(k)) during silent intervals and this information is later used to reconstruct the silence segments in the receiver using simulated noise of the correct amplitude and duration.

Figure 4 shows the design of the simulated noise source which uses a binary shift-register generator corresponding to the primitive polynomial $h(x) = X^{18}+X^7+1$ to generate a psuedo-random binary sequence [4]. The repetition period of the shift-register output using this polynomial is $2^{18}-1$, which corresponds to a 33 second repetition rate at the 8000 Hz sampling rate. This is long enough to prevent an audible cadence to be detectable in the pseudo-random noise. A practical advantage of this polynomial is the minimum number of feedback taps (2) required which allows this polynomial to be very efficiently programmed.

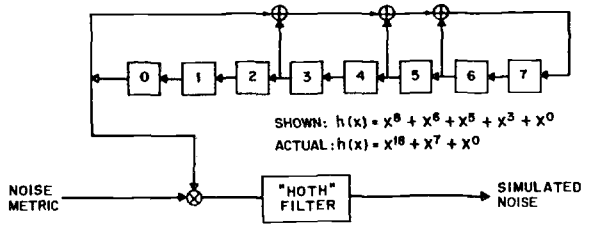

Fig. 4 Simulated Background Noise Source

The binary sequence thus formed is then used as the modulating signal to generate a spectrally flat (white) noise at the amplitude of the original noise segment detected at the transmitter as shown in figure 4. The white noise product is passed through a low pass filter which shapes the simulated noise to approximate the average coloration of the background noise [2].

$$H(z) = \frac{.325}{1-.675z^{-1}} \tag{12}$$

It is also necessary to guard against audible switching transients when the receiver switches between speech decoding of speech segments and simulated noise generation during silence segments. Two techniques were used to prevent these "clicks": The first is to switch the speech decoders input to a dummy data stream that flushes the decoders buffers to zero, thus preventing a mismatch between the end of a speech segment and the beginning of the next speech segment. In effect, the speech decoder is left operating and only its input is switched to a dummy data stream during silence segments. The second technique uses a linear switch to add the simulated noise to the decoder output thereby smoothing the transition between segments. The equations for performing this function are:

$$r(k) = (1-B_{sw})b(k) + B_{sw}r(k) \tag{13}$$
$$o(k) = d(k) + r(k)p(k) \tag{14}$$

In equation 13 b(k) is a binary function that equals 1.0 during silence segments and 0.0 during speech segments. The constant B_{sw} is the switching time constant and is set to .975. The simulated background noise is r(k), d(k) is the speech decoder output, and o(k) is the decoder output containing decoded speech and simulated noise.

V. Performance

The silence compression technique described has been programmed with a speech coding algorithm [5] into a single DSP-20 signal processor and informal tests were performed by the authors to determine the quality and intelligibility, the average overall data compression, and the robustness of this system to various noise environments. The results of these preliminary evaluations are given below.

This silence compression has no effect on intelligibility and has little effect on perceived quality given that the speech signal to noise ratio (SNR) of the source is at least 20 dB. For inputs with SNR values between 10 dB and 20 dB, silence compression does not significantly degraded intelligibility but the perceived quality is degraded due to the "chopping off" of low level sounds at the ends of speech segments. For inputs with SNR values below 10 dB the intelligibility is also degraded due to misclassification of speech as noise.

The percentage of data reduction silence compression offers is directly proportional to the percentage of silence present in the source. Our evaluations indicate a data reduction of 20% to over 50% depending on the talker and subject. One author [3] has provided statistics on the average percentage of "pauses in isolation" during the naturally spoken conversations of 16 subjects which indicate that subjects pause 30% of the time during "uninterrupted" speech on the average. This data combined with our tests imply at least a 20% data reduction can be expected using this silence compression technique.

Finally we evaluate the effect of nonstationary background noises on silence compression. Nonstationary background sounds can be divided in threes classes for this purpose:

1. Noise levels that quickly decrease to a new stationary level.
2. Noise levels that quickly increase to a new stationary level.
3. Noise levels that have no stationary level.

If the noise level suddenly decreases (eg: a fan is turned off) this system instantly adapts to the new lower noise level (equation 8) and continues to operate. Conversely, if the noise suddenly increases in level (eg: a fan is turned on) then approximately 5 seconds is required to adjust to the new level (equation 7) during which time the noise energy will be classified as speech. In cases of highly nonstationary background noise (eg: a typewriter) the noise spikes will be classed as speech and silence compression will be minimized.

VI. Concluding Remarks

The goal of this work was to create a silence compression algorithm combining computational efficiency, robustness, and good subjective quality. The technique described in this paper has met these objectives. The only caveat is that in endeavors such as this where performance is traded against complexity, some decisions must usually be made which are application dependent.

REFERENCES

[1] P. T. Brady, "A Technique for Investigating the On-Off Patterns of Speech", BSTJ, Jan., 1965, p.1.

[2] Daniel F. Hoth, "Room Noise Spectra at Subscribers Telephone Locations," J.A.S.A., April 1941, Vol. 12, p499.

[3] P. T. Brady, "A Statistical Analysis of On-Off Patterns in 16 Conversation", BSTJ, Vol 47, January 1968, p87.

[4] W. Wesley Peterson, E. J. Weldon, Jr., "Error-Correcting Codes," The MIT Press, Second edition, p476.

[5] R. E. Crochiere, R. V. Cox, J. D. Johnston, "Real-Time Speech Coding," IEEE Transactions on Communications, V.30 No.4, April 1982, p621.

[6] J. L. Flanagan, "Digital Voice Storage in a Microprocessor," IEEE Trans. on Comm., Vol.30, No.2, February 1982.

[7] M. B. DonVito, "Subband Coding with Silence Detection," ICASSP 85 Proceedings, Vol.4, p1433.