Voice Activity Detection based on Wavelet Transform

Pranav Venuprasad
A53241169
(Teammate : Jacob T Lassen)

***Abstract*—In this paper, we look at 2 different methods for Voice Activity Detection (VAD) based on the Wavelet Transform. The algorithms utilize the Wavelet Transform's flexibility in the time-frequency resolution to compute robust parameters for VAD decision. We examine the performance of the algorithms in clean and various noisy environments with varying levels of noise.**

I. Introduction

Voice Activity Detectors are common algorithms in digital speech processing and is used in a wide variety of applications as a pre-processing step. In speech coding, it is used to reduce amount of transmitted data, by switching off the transmission when there is no speech. In speech recognition, VAD saves processing power by sending only the parts with speech to the recognition engine. It can also be used to detect background noise, and then minimize the background noise from the speech signal. The basic task here is to find the regions of an audio signal which contain speech content. Figure 1 illustrates a voice activity detector in action on a an audio signal containing speech.

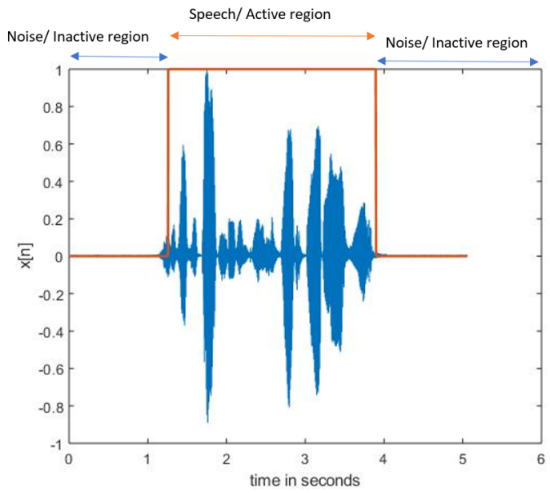

Fig. 1. Illustration of VAD on a speech signal.

Common VAD algorithms use decision parameters that are based on averages of temporal parameters such as autocorrelation coefficients, zero-crossing rate or short term energy. These methods are simple and have low computation cost, but their effects are not perfect and generally only suitable for the situation with high signal-to-noise ratio (SNR). Here, we explore algorithms based on the Wavelet transform which give reasonable performance in even noisy conditions.

II. Discrete Wavelet Transform

One of the most commonly used frequency domain features is the Short Time Fourier Transform (STFT). The STFT has the same time resolution for all frequency bands. However in many practical applications, it would be beneficial to have a variable time resolution for different frequency bands. For this purpose, the continuous wavelet transform was designed and it is defined as:

$$\gamma(s,\tau) = \int_{-\infty}^{\infty} f(t)\psi^*_{s,\tau}(t)dt$$

where the wavelets $\psi_{s,\tau}(t)$ are generated from a mother wavelet $\psi(t)$ as :

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-\tau}{s})$$

The resulting coefficients are functions of scale and position. A low scale (detail part) indicates a compressed wavelet which detects rapidly changing details, which usually contains most of the speech content, whereas a high scale (approximation part) stretched the wavelet, showing low frequencies or noise. In this paper, we use the Discrete Wavelet Transform (DWT) and denote the detail coefficients as $d(n)$ and approximation coefficients as $a(n)$.

III. Algorithm

A. Method 1

Assuming that $s(n)$ is a clear speech, $w(n)$ is an additive noise. The speech with noise can be written as:

$$y(n) = s(n) + w(n)$$

Taking discrete wavelet transform of both sides, we have:

$$d_y(n) = d_s(n) + d_w(n)$$
$$a_y(n) = a_s(n) + a_w(n)$$

where $a_y(n), a_s(n), a_w(n)$ denote the approximation component of noisy speech, clear speech and noise respectively, $d_y(n), d_s(n), d_w(n)$ denote their detailed

coefficients respectively. At a large scale, the detail component of a noisy speech is mainly determined by speech. The amplitude of detail component of a noise is usually very small. Therefore, the detail components of noisy speech and noise are quite different. Straightforward, at a proper scale, the average energy of noisy speech is greater than the one of noise. In other words, their relationship can be written as:

$$\frac{1}{M}\Sigma_{n=1}^{M}[d_y^{(j)}(n)]^2 > \alpha\frac{1}{M}\Sigma_{n=1}^{M}[d_w^{(j)}(n)]^2, \alpha > 1$$

where M denotes the frame length of speech, j denotes the scale of wavelet transform, α denotes an experiential coefficient. To make the result more reliable, two scale j = 3, 4 detail components are used. Here we assume that the initial 4 frames of the signal are noise and the root-mean-square (RMS) of detail components of noise is initially computed by the first four frames. The steps of the VAD algorithms are given as follows:

Step 1: Initialize : Assume at the first 4 frames of $y[n]$ are noise and compute the RMS of the detailed components $\bar{d}_w^j(j = 3, 4)$ as

$$\bar{d}_w^j = \frac{1}{4M}\Sigma_{n=1}^{4M}[d_w^{(j)}(n)]^2 \ .$$

Step 2: The current frame data are saved in $y[n]$. Calculate the RMS of detail components $\bar{d}_y^j(j = 3, 4)$ as

$$\bar{d}_y^j = \frac{1}{M}\Sigma_{n=1}^{M}[d_y^{(j)}(n)]^2.$$

Step 3: Carry out VAD based on the following equation

$$VAD_{out} = \begin{cases} 1, & \text{if } \bar{d}_y^3 + \bar{d}_y^4 > \alpha\bar{d}_w^3 + \beta\bar{d}_w^4. \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Step 4: Refresh data. If a speech is present, then $w[n]$ is held; otherwise, the first frame of $w[n]$ is as output, and current frame($y[n]$) is put into $w[n]$.

Step 5: Go to step 2 and repeat till end of signal is got to.

In this paper, we choose the values of the parameters as $\alpha = \beta = 1.8$ and use the Daubechie wavelet ($dB3$) for the Discrete Wavelet Transform.

B. Method 2

The second method we investigate is based on [2] and consists of estimating 4 different energy-based parameters from the wavelet coefficients to classify a region as either *silence, stationarity* or *background noise* using 4 fixed thresholds. First the input speech signal is segmented into frames of length M. Then 4 different binary flags is set for each segmented frame: f_{sil} for *silence*, f_{stat} for *stationarity*, and f_{B2} and F_{BL} for *background noise*. These four flags will then be used to compute a VAD decision by

$$\text{VAD} =! \left(f_{sil}\right| \left(f_{B2}\&f_{BL}\&f_{stat}\right)) \tag{2}$$

where '!, |, &' are the logical operators 'not, or, and'. The VAD decision is made by saying the frame is not speech if it is classified as silence or stationary background noise.

1) Silence detection: First the frame energies $E_1, ..., E_L$ of the detail coefficients $d^{(1)}(n), ..., d^{(L)}(n)$ and the frame energy E_{L+1} of the approximation coefficients $a^{(L)}(n)$ is computed. Then the binary flag can be set if the total frame energy $E_{tot} = \sum_{l=1}^{L+1} E_l$ is smaller than the fixed threshold T_1.

$$f_{sil} = \begin{cases} 1, & \text{if } E_{tot} < T_1. \\ 0, & \text{otherwise.} \end{cases}$$

2) Stationarity detection: The stationarity decision is made by first finding the k-th frame difference measure from the energy of all the detail coefficients

$$\Delta^{(k)} = \sqrt{\frac{1}{L}\sum_{l=1}^{L}\left(E_l^{(k)} - E_l^{(k-1)}\right)^2}.$$

The flag for *stationarity* is then set based on the current and previous frame if the difference measure is below threshold T_2 for both

$$f_{stat} = \begin{cases} 1, & \text{if } \left(\Delta^{(k)} < T2\right) \& \left(\Delta^{(k-1)} < T_2\right). \\ 0, & \text{otherwise.} \end{cases}$$

3) Background noise detection: The energy of the detail coefficients at two different levels are used in order to set two flags that determines whether the current frame is background noise or not. The energy of the detail coefficients $d^{(L)}(n)$ is computed for each frame while the energy of the detail coefficients $d^{(2)}(n)$ are computed for each subframe. The subframes are found by segmenting each frame into P subframes of length M/P.

The current background noise level $B_i^{(k)}$, $i \in \{2, L\}$ is estimated as

$$B_i^{(k)} = \begin{cases} E_i^{(k)}, & \text{if } B_i^{(k-1)} > E_i^{(k)}. \\ \alpha B_i^{(k+1)} + (1-\alpha)E_i^{(k)}, & \text{otherwise.} \end{cases}$$
$$0 < \alpha < 1$$

Then the P subframe energies $\epsilon_2^{(k,1)}, ..., \epsilon_2^{(k,P)}$ are computed for $d^{(2)}(n)$ and the two flags at different detail levels are set as

$$f_{B2} \begin{cases} 1, & \text{if } [(\epsilon_2^{(k,1)} - B_2^{(k)}) < T_3] \& \cdots \& \\ & \quad [(\epsilon_2^{(k,P)} - B_2^{(k)}) < T_3]. \\ 0, & \text{otherwise.} \end{cases}$$

$$f_{BL} = \begin{cases} 1, & \text{if } (E_L^{(k)} - B_L(k)) < T_4. \\ 0, & \text{otherwise.} \end{cases}$$

In order to get good VAD decision the 4 thresholds T_1, T_2, T_3, and T_4 along with α must be chosen carefully. For the Discrete Wavelet Transform the Daubechie wavelet ($dB4$) is used.

C. Runlength Filtering

We observed that the output VAD decision sequence has multiple transitions near the speech/noise boundaries. Ideally, we would like to get a single transition near each boundary and for this purpose, we have implemented a run-length filter that filters the decision sequence. Here we have 2 parameters : N_{in} and N_{out}. When transiting from a noise to speech section, the transition in the VAD from 0 (speech absent) to 1(speech present) is detected only when atleast N_{in} consecutive speech frames are detected. Similarly when transiting from a speech to a noise section, the transition in the VAD from 1 to 0 is detected only when N_{out} consecutive noise frames are detected. Note that this run-length algorithm was not dealt with in the references and we have incorporates it from our side. Here, we use use $N_{in} = N_{out} = 4$ frames, corresponding to a threshold of 120ms. The effect of the run-length filter is illustrated in Figure 2 below.

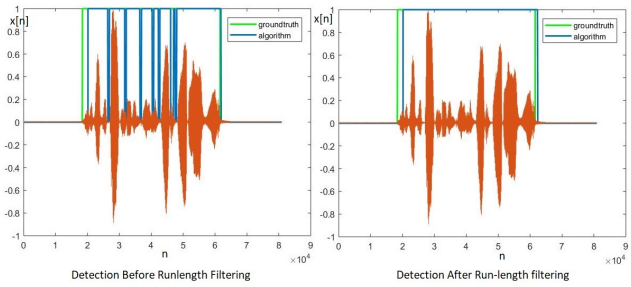

Fig. 2. Illustration of Runlength filtering.

IV. Experiments

The VAD algorithms are implemented on the speech files from the GRID dataset [3]. The test is performed on 16 sentences spoken by 8 males and 8 females. The signals are manually corrupted using 4 different classes of noise from the NOISEX database [4]: white noise, factory noise, cockpit noise and background speech noise. The noise signals are added scaled for the SNR levels : 10dB, 20dB, 30dB, 40dB, 50dB and ∞. Note that a low SNR value corresponds to a highly corrupted signal and a high SNR represents a less noisy signal. SNR=∞ corresponds to clear speech. Here, we use frames that are 30ms long, with no overlap.

For Method 2, $P = 2$, and $L = 4$ are chosen as per [2] and the thresholds are set as follows: T_1=1e-10, T_2=1e-4, T_3=1e-5, and T_4=1e-5 along with α=0.5. They were chosen through trial-and-error as [2] did not give the exact values in the paper.

A. Evaluation Metrics

For evaluation, we define the following metrics:

1) Speech as Noise (SAN): Speech as Noise is defined as the percentage of actual speech frames that are classified as noise by the algorithm.

2) Noise as Speech(NAS): Noise as Speech is defined as the percentage of actual noise frames that are classified as speech by the algorithm.

3) Percentage Activity: Percentage Activity is defined as the percentage of total number of frames that are classified as speech by the algorithm. The obtained values of Percentage Activity are compared with the groundtruth values, i.e. the actual percentage of speech frames.

4) Clipping: The clipping rate is defined as the percentage of frames that are classified as noise by the VAD and had been previously rated as active for non-noisy speech.

V. Results

Figures 3 and 4 show the results of the algorithms on an audio signal with added factory noise at various SNRs.

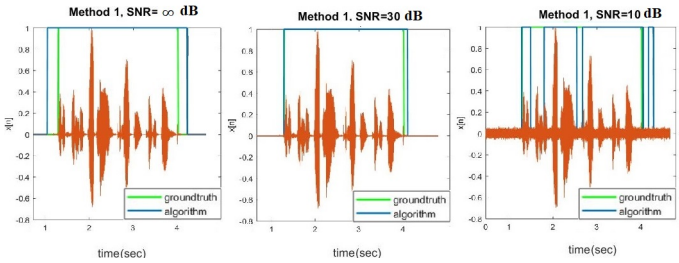

Fig. 3. Method 1 with factory noise.

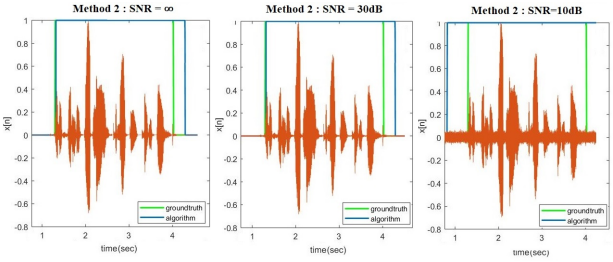

Fig. 4. Method 2 with factory noise.

Figure 5-8 show the performance of the algorithms for different kinds of noise at varying SNRs averaged across 16 audio signals.

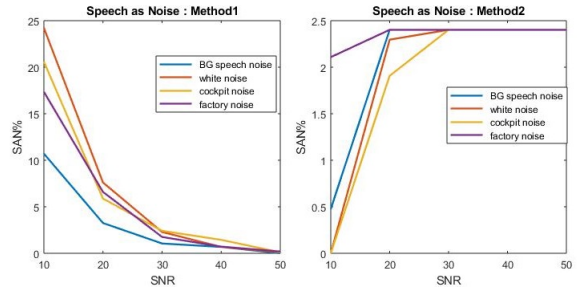

Fig. 5. Results: Speech as Noise.

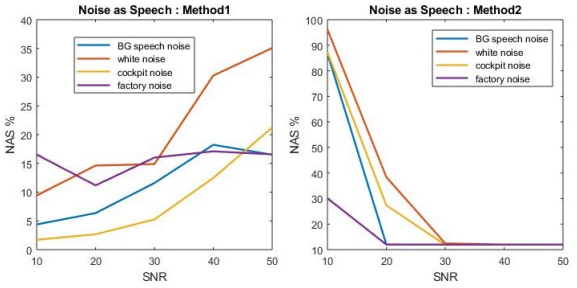

Fig. 6. Results: Noise as Speech.

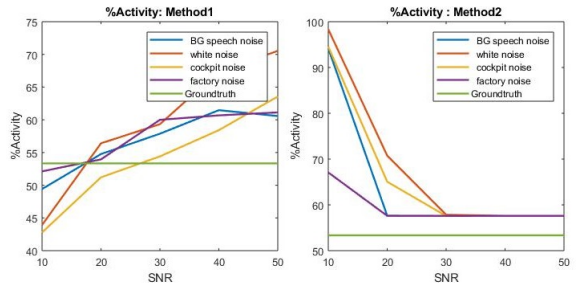

Fig. 7. Results: Percentage Activity.

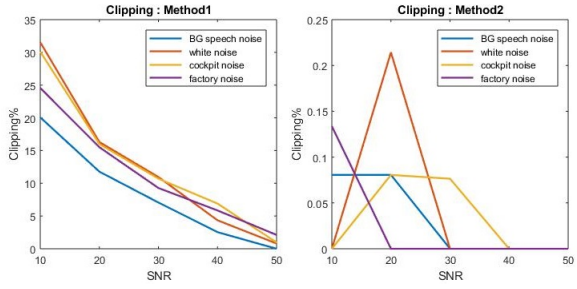

Fig. 8. Results: Clipping.

VI. Discussion

Starting with the effect of the Runlength filtering, from figure 2, we see that the filtering is a crucial post-processing task. Without the filtering, the number of speech/noise transitions will be very high, which is not suitable for the practical applications of VAD.

From figure 3, we observe that Method 1 actually performs better in the presence of a small noise compared to clean speech. In the SNR=∞ case, we can notice a chunk of frames before the start and after the end of the actual speech, being classified as active, which is not the case for SNR = 30dB. This is because the threshold for noise energy is adaptive, and the threshold is better modelled in the presence of a small noise. However, when the noise increases so as to be comparable to the actual signal, then many of the actual speech frames are missed in the detection.

From figure 4, we see that the Method 2 has a chunk of frames after the end of the actual speech being classified as active. But performance of Method 2 does not change much with the addition of a small noise. However, when a large noise is added, then it tends to classify all frames as active.

Hence we observe some kind of trade-off between the performance of the 2 methods when a large noise is added. For low SNR, method 1 tends to speech frames as noise and method 2 tends to classify noise frames as speech. The same trend is observed from the comparison graphs (fig 5-8) as well. Note that the graphs here are not to scale. Comparing the SANs in figure 5, we see that for high SNR, neither of the 2 methods miss out in identifying the speech frames. However as SNR decreases, method 1 tends to drop more speech frames, but method 2 doesnot. Hence in applications where we are concerned about speech being dropped, method 2 works better. As for NAS, we observe from Figure 2 that both methods classify some part of the noise as speech in the case of clean speech. As SNR decreases, almost all noise frames are classified as speech by method 2, whereas method 1 actually produces lesser false positives.

Method 2 always gives a higher average percentage activity than the actual value, with the percentage going as high as 100% for low SNR, whereas method 1 floats around the actual value. So in cases where the bandwidth available to send the signal is limited, method 1 works better.

As for clipping, since method 2 is only concerned about classifiying speech as noise, we see that the clipping is approximately zero. This is because as SNR decreases, more noise frames are classified as speech and a frame classified as speech in the SNR= ∞ case will always be classified as noise. But method 1 gives increased clipping as SNR decreases.

Hence from the discussion above we can say in general that method 1 performs better than method 2 in noisy conditions. Note that the 10dB case represents a very high noise that is impractical (considered for completeness of analysis) and SNR=30dB is a more practical value. Method 2 performs better in the case of

clear speech/ low noise conditions. Also, in applications concerned about the speech frames being dropped, like mobile phone lines, method 2 works better. However, if we are more concerned about the average bit rate and the available bandwidth is limited, method 1 can be employed.

VII. References

[1] J. Shaojun, G. Haitao and Y.Fuliang, "A new algorithm for voice activity detection based on Wavelet Transform"

[2] J.Stegmann, G.Schroder,"Robust voice activity detection based on the wavelet transform"

[3] M. Cooke and J. Barker, "An audio-vidual corpus for speech perception and automatics speech recognition" : (GRID database)

[4] A. Varga and H. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems"

[5] ETSI: "Draft Recommendation prETS 300 724: -GSM Enhanced Full Rate (EFR)speech codec", 1996.