A Comparison of Acoustic Parameters and MFCCs for Speaker Identification

Alec Colvin, Srikanth Vishnubhotla,
Daniel Garcia-Romero, and Carol Espy-Wilson

1 Purpose

The field of computer-based speaker identification has been dominated by the use of Mel-Frequency Cepstral Coefficients (MFCCs) -- a set of a dozen or more coefficients created by projecting the logarithmically sampled audio cepstrum into a DCT basis. MFCCs are the long-standing favorite because they work well and are computationally efficient, but they aren't derived from an understanding of the speaker's characteristic qualities. They were designed to describe the linguistic content of the speech signal, and it is only incidental that they capture speaker-specific information as well.

This investigation was motivated by the study performed in 2006 by Srikanth Vishnubhotla and Professor Espy-Wilson that investigated the use of acoustic parameters (APs) for text-independent speaker identification, rather than the conventional use of MFCCs.[1] The acoustic parameters were first used as a speaker ID tool by the University of Maryland Speech Communication Lab in an attempt to intuitively characterize the intrinsic qualities of a speaker. Intrinsic qualities include the amount of closure of the vocal chords, referred to as voicing, and the location of the smallest constriction of the vocal tract, which varies from one speaker to another, even for the same sounds. Of the eight APs, the four formants describe the speaker's use of his vocal tract, the periodic and aperiodic energy describes the speaker's phonation, and the voice quality -- either creaky, modal, or breathy -- is described by the amplitude difference of the first two harmonics (H1 minus H2) as well as the spectral slope. Creakiness occurs when the glottis is pressed during vocalization and can best be described as the grating sound of a chain smoker or somebody who just woke up with a parched throat; modal is synonymous with normal speech; and breathy occurs when glottis doesn't close completely during vocalization. Breathy is the hardest to describe, but Marilyn Monroe's rendition of "Happy Birthday" is an extreme example. The study of these eight parameters was largely successful because it showed that the 8 APs performed comparably to a set of 39 MFCCs.

The 2006 study was performed using the NIST '98 Evaluation Database, which is telephone speech sampled at 8 kHz, and therein lies the impetus of this summer's work. Telephone speech is degraded during the digitizing process, but this is hardly noticeable because the poor quality of handset microphones and earpieces distorts the audio signal much more than any other part of the telephone system. Nonetheless, the DC notch filter in the NIST '98 Evaluation Database recordings is unacceptable for a thorough evaluation of the acoustic parameters because the notch filter creates attenuation in the first harmonic, which interferes with the computation of H1-H2.

The Buckeye Corpus is a collection of hour-long interviews with Ohio natives near in and around Columbus, Ohio. The corpus creators at Ohio State University realized the limitations of the telephone databases and chose to record the interviews with a microphone. The result is a nearly pristine set of data sampled at 16 kHz (rather than the 8 kHz that telephones use). There is a slight hum at 60 Hz due to a poorly-grounded power source, but it hovers naer -40 dB, which is never enough to be statistically significant. The 16 kHz sampling rate is particularly useful because it means that the Buckeye Corpus can be used to study the discriminative power of the fourth and fifth formants (which characterize the vocal tract of the speaker), and it is somewhat useful in this study since we chose to use four formants and the fourth formant is more prominent in an audio sample with an 8 kHz bandwidth.

In short, the objective of this study was to repeat the procedure of the study done last year to show whether or not better results could be obtained by using the high-fidelity recordings of the Buckeye Corpus. Since the H1-H2 parameter is the only one of the eight acoustic parameters directly affected by the increased quality of the Buckeye Corpus, this study was also planned as an investigation of the H1-H2 parameter.

2 Procedure

2.1 Extract the Voiced Phones

There are many complex ways to phonetically analyze a sentence, but one of the simplest methods is the division into voiced sounds, unvoiced sounds, and silence. Of the eight acoustic parameters, only it is just the H1-H2 parameter that doesn't produce meaningful results in the unvoiced and silent regions because it assumes a harmonic structure (a vibration of the vocal chords) that only occurs in voiced sounds. The algorithms that have been developed for the formants and the H1-H2 will produce values for the entire audio file, but the values in the unvoiced and silent regions skew the models in unpredictable and meaningless ways, so it was decided to use the detailed phonetic transcription associated with each interview recording to surgically remove all of the unvoiced and silent regions of the recordings. While we cannot claim that the results obtained in this study are valid for conversational speech, we can say that the results are more meaningful because we are not using spurious output from the formant and H1-H2 algorithms.

2.2 Compute the Formant Values

The formant values were computed using the Wavesurfer application, which computes formants using the ESPS Toolkit. Default values were used for all of the parameters, such as the FFT size and the window size, and formants were computed for every 10 milliseconds of data. An example is shown in Figure 1 which shows a spectrogram of speaker 1, file 1 from the Buckeye Corpus with four formant tracks as calculated with Wavesurfer.

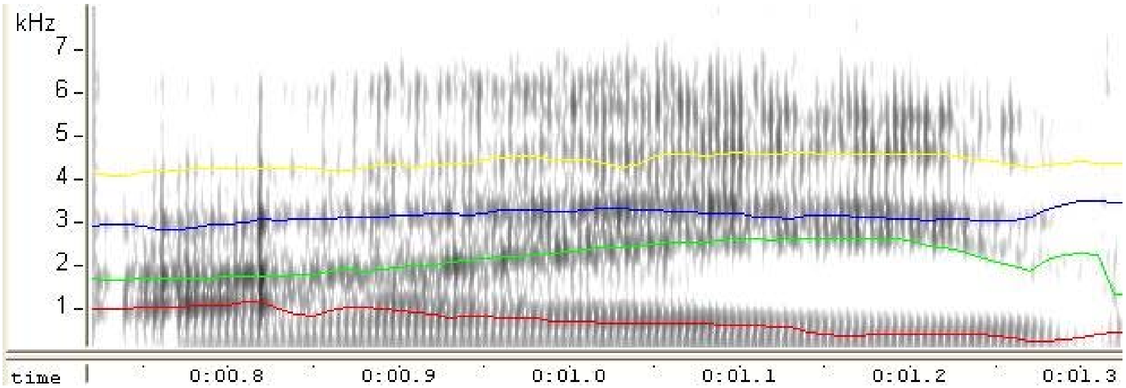


Figure 1: spectrogram of speaker 1, clip 1 of the Buckeye Corpus with the four formants overlaid

2.3 Compute the H1-H2 Values

Finding the H1-H2 values is a six-step process:

1. Determine whether the current 40 millisecond window is creaky or modal by using the log files associated with each waveform file.
2. If the current window is creaky then compute the energy spectrum of the window. Otherwise, compute the glottal spectrum of the window.
3. Identify all of the local maxima (peaks) of the spectrum.
4. Identify the largest peak between 60 Hz and 300 Hz. This is the frequency of the speaker's pitch at the middle of the time window, referred to as F0.
5. Find a spectral peak near 2*F0. This is the frequency of the second harmonic.
6. Report H1-H2 as the difference in amplitude of the first and second harmonics.

Because it uses as 40 millisecond window, the algorithm for computing H1-H2 isn't valid for the first 20 milliseconds or the last 20 milliseconds, so the results need to be padded with two zeros at the beginning and the end of the results in order for the H1-H2 values to be aligned with the other 7 APs. The result of the H1-H2 computation for a single time window is seen in Figure 2.

2.4 Compute the Spectral Slope

The spectral slope is computed by fitting a straight line to the first 1200 Hz of the spectrum using Matlab's *polyfit* function. We only used the first 1200 Hz because the aspiration due to a breathy voice can skew the results. Figure 2 shows the physical configuration of the glottis for each of the three modes, as well as the waveform and the spectrum. In the spectrum plots, you can see that the H1-H2 values and the spectral slope are both sensitive to the mode.

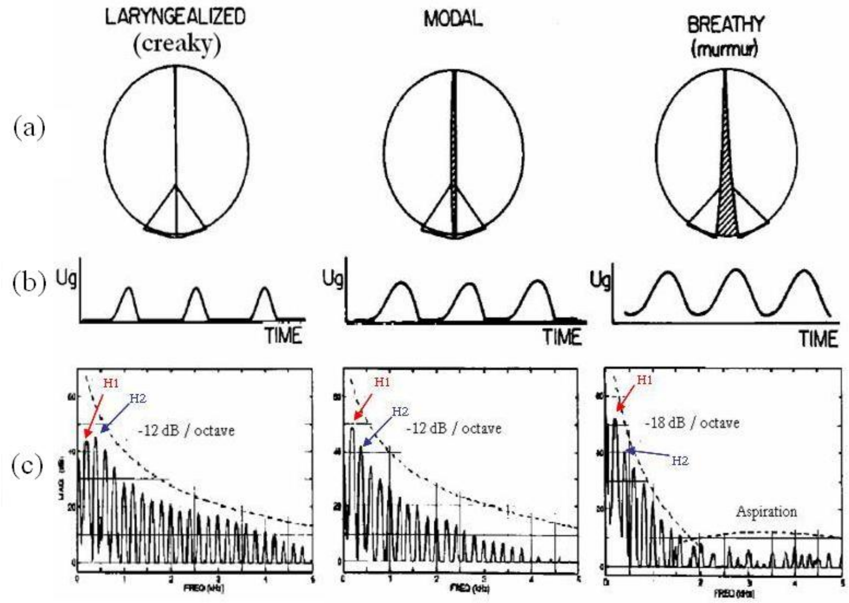


Figure 2: schematics of (a) vocal chords, (b) resulting glottal waveform, and (c) glottal spectrum [adapted from Klatt & Klatt, 1990] [2]

2.5 Compute the Periodic and Aperiodic Energy

Periodic and aperiodic energy is measured using the APP Detector, developed by Om Deshmukh as part of his master's thesis.[3] The APP Detector operates by computing the Average Magnitude Difference Function (AMDF) for the current time window, and measuring the magnitude and frequency of the AMDF minima.[4] Strong minima at regular intervals tells us that periodic energy is present; weak minima at random intervals tells us that aperiodic energy is present. As a general rule, creaky voices exhibit large aperiodic energy while modal and breathy voices exhibit periodic energy.

2.6 Compute the MFCCs

It is customary to use 13 MFCCs for a corpus sampled at 8 kHz. Researchers will often use the first and second derivatives of the MFCCs, resulting in the 39 data points that were used in last year's study. The Buckeye Corpus' 16 kHz sampling rate is a bit unusual, so we felt that the 13 MFCCs were no longer appropriate. Therefore we chose to use 30 MFCCs as well as the time derivatives of each of the 30 MFCCs, resulting in 60 data points every 10 milliseconds.

Computation of the MFCCs is performed using the *gen_feat* program provided by the MIT Lincoln Lab. In addition to the program defaults, we chose to use Rasta filtering, a technique developed by Hermansky which has proven to be very effective at removing noise and channel artifacts.[5]

2.7 Generate 2½ Minute Feature Files

Once all of the data files had been computed, they needed to be arranged into data files representing 2.5 minutes worth of speech. A file length of 2.5 minutes is an arbitrary trade-off. On one hand, the files should be long because they will be used to train the Universal Background Model, and longer training sequences yield better models. On the other hand, the files should be short because they will be used as test cases against other models, and short test cases are representative of a short telephone conversation.

2.8 Train the Universal Background Model

Using the procedure described by Reynolds, a Universal Background Model was trained using the entire database in order to provide prior knowledge for the individual speaker specific models.[6]

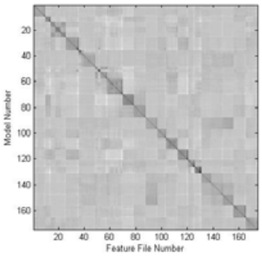
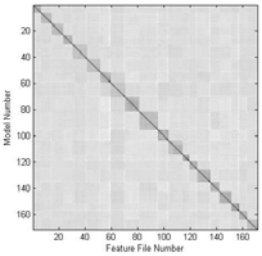
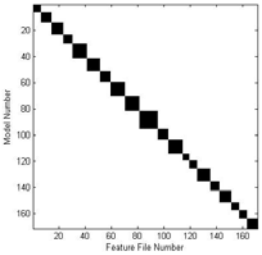
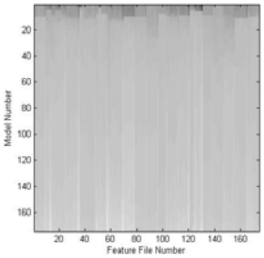
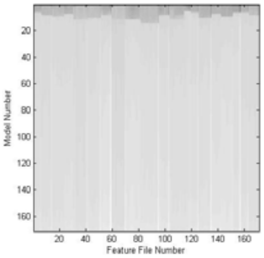
2.9 Compute the Results and the Percent Accuracy

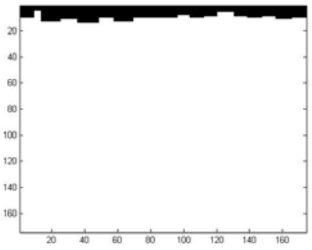
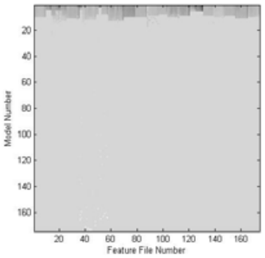
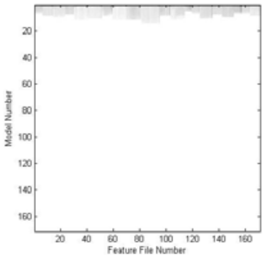
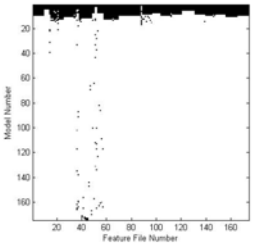
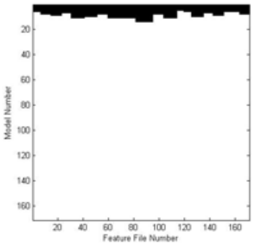
Each of the 171 female models were compared against the 171 feature files, producing 29241 scores. The same was done for the 174 male models, producing 30276 scores. For both the male and female models, there is a six-step procedure:

1. Compile the scores into a two-dimensional matrix, with the columns representing the feature file and the rows representing the model number.

2. Create a mask matrix representing the correct answers. In other words, if row 3 is based off of speaker 1 and columns 3 is based off of speaker 1, then mask(3,3) will be 1. If a given row and column are based off of different speakers then the matrix entry mask(x, y) will be 0.
3. Sort each column of the results in a descending manner.
4. Sort each column of the mask in a descending manner.
5. Multiply the mask by the results.
6. Look at every element in the results. If an element is not zero then set it to 1.

The plots for each step are given below. Note that the lowest number is always represented as white and the highest number is always represented as black, but the dominant shade of gray changes from between steps because the data is being altered.

Step Number	Mask	Imperfect Results	100% Accurate Results
1			
2			
3			

Step Number	Mask	Imperfect Results	100% Accurate Results
4			
5			
6			

From the plots in step 6 it can be seen that every error results in two incorrect pixels: a black pixel where there should be a white pixel, and a white pixel where there should be a black pixel. Essentially, a black pixel trades places with a white pixel every time an error occurs. Therefore, the number of computational errors is exactly half the number of incorrect pixels, and the percent error is the number or errors divided by the number of black pixels. From there, the percent accuracy is simply (1 - percent error).

3 Results and Conclusions

% Accuracy	Male	Female
MFCC	99.67	100.00
7 APs (no H1-H2)	95.72	99.23
8 APs	94.17	98.21

Initially, the second row of the results table was not computed, but when we saw that the APs performed worse than the MFCCs we computed the second row to see what effect the H1-H2 parameter has on the overall performance.

Even though the H1-H2 parameter was detrimental to the overall performance, we believe that it might prove useful if its reliability can be improved. An easy way to improve the reliability of H1-H2 is to improve the detection of creaky regions. Creakiness was only labeled in the Buckeye database in places that were noticeable creaky, but creakiness is often present in a statistically significant way even when it is not audibly present. If we choose to treat all of the aperiodic regions identified by the APP Detector as creaky, then we can have a much better idea of when to use the glottal spectrum during the calculation of H1-H2. Another approach to improving H1-H2 reliability is to eliminate cases of pitch halving and pitch doubling. In both of these improvements are made then we believe that the H1-H2 parameter will help us to match the performance of the MFCCs.

[1] Espy-Wilson, C., Manocha, S. and Vishnubhotla. S., "A new set of features for text-independent speaker identification," International Conference on Spoken Language Processing, Interspeech 2006.

[2] D. Klatt and L. Klatt, “Analysis, synthesis and perception of voice quality variations among female and male talkers,” J. Acoust. Soc. Am. 87, 820-857, 1990.

[3] O. Deshmukh, "Synergy of Acoustic-Phonetics and Auditory Modeling Towards Robust Speech Recognition," Ph.D. Thesis, 2006.

[4] O. Deshmukh, C.Espy-Wilson, A. Salomon & J. Singh, “Use of Temporal Information: Detection of the Periodicity, Aperiodicity and Pitch in Speech," IEEE Trans. on Speech

[5] H. Hermansky and N. Morgan, "RASTA processing of speech", IEEE Trans. on Speech and Audio Proc., vol. 2, no. 4, pp. 578-589, Oct. 1994.

[6] Douglas A. Reynolds, Thomas F. Quatieri, Robert B. Dunn (2000): "Speaker verification using adapted Gaussian mixture models", Digital Signal Processing, vol. 10, no. 1-3, Jan-July 2000