WAVELET BASED SPEECH RECOGNITION

Brian Gamulkiewicz and Michael Weeks

Computer Science Department,
Georgia State University
Atlanta, Georgia, 30303, USA

ABSTRACT

The problem of speech recognition is addressed using the wavelet transform as a means to help match phonemes from a speech signal. This work uses a template of pre-recorded, wavelet-transformed phonemes as its basis for comparison. This application illustrates how wavelets can be used for better accuracy in speech recognition [1].

1. INTRODUCTION

Speech recognition is currently used in many real-time applications, such as cellular telephones, computers, and security systems. However, these systems are far from perfect in correctly classifying human speech into words. Speech recognizers consist of a feature extraction stage and a classification stage. The parameters from the feature extraction stage are compared in some form to parameters extracted from signals stored in a database or template. The parameters could be fed to a neural network or hidden Markov model as well [1].

In this paper we use the wavelet transform with a voice recognition system. Because speech analysis is a very complicated task, the analysis is kept as simple as possible while trying to arrive at the correct conclusion. By analyzing and transforming the speech signal, a specific value can be calculated to determine the phonemes said by the user. The goal of this paper is to develop a speech recognition algorithm that uses the wavelet transform to extract and represent incoming speech signals as a basis for an accurate method of identifying and matching these signals to signals in a template.

Section 2 discusses the speech recognition background. Next, section 3 details our approach. Section 4 presents the results, and section 5 contains our conclusions.

2. BACKGROUND

Problems in recognizing speech include noise, speaker variations, and differences between the training and testing environments, such as the microphones used [2]. One way of dealing with this is to adapt the recognition system's internal model (i.e. Hidden Markov Model weights). Another is to normalize the new speech to conform with the training data. Variations with different speakers mean that speaker-dependent systems usually do better than speaker-independent ones, since the former uses the speaker for training.

Dynamic Time Warping, or a similar algorithm, is necessary because of the non-uniform patterns of different speech signals. Also, different speakers will more than likely say the same words at different rates. This means that a simple linear time alignment comparison, such as the root square mean error, cannot be used efficiently.

One way to do speech recognition is phoneme-based indexing [3]. A phoneme is a basic sound in a language, and words are made by putting phonemes together. One method is to consider the triphone, a set of three phonemes where a phoneme is considered with its left and right neighbors [4]. Therefore, this method identifies speech based on its component phonemes.

We are not trying to match a spoken word to a word list, but rather output the phonemes detected. For example, if the user says the word "pocket", our system should output "p", "ah", "k", "eh", "t".

Our approach includes the wavelet transform, shown in figure 1 [5]. This figure shows that a 1-dimensional signal is broken into two signals by low-pass and high-pass filters. The downsamplers (shown as an arrow next to the number 2) eliminate every other sample, so that the two remaining signals are approximately half the size of the original. As this figure shows, the low-pass (approximate) signal can be further decomposed, giving a second level of resolution (called an octave). The number of possible octaves is limited by the size of the original signal, though a number of octaves between 3 and 6 is common.

Wavelets express signals as sums of wavelets and their dilations and translations. They act in a similar way as Fourier analysis but can approximate signals which contain both large and small features, as well as sharp spikes and discontinuities. This is due to the fact that wavelets do not use a fixed time-frequency window. The underlying principle of wavelets is to analyze according to scale.

This work was supported by State of Georgia's Yamacraw project.

[1] This paper was published in the *IEEE Midwest Symposium on Circuits and Systems (MWSCAS)*, Cairo, Egypt, December 27-30, 2003.

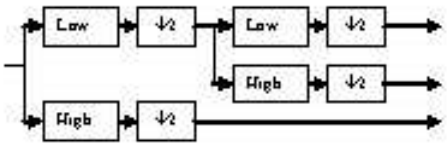

Fig. 1. The Discrete Wavelet Transform

3. OUR APPROACH

We are doing phoneme matching, for a speaker dependent system. The algorithm to classify speech into words is as follows:

- Read-in signal for analyzing and matching
- Eliminate any silence of the signal
- Normalize signal around x-axis
- Normalize amplitude values
- Use wavelet transform (Daubechies 8) to obtain five octaves of the same signal
- Compare this signal to template by calculating errors between them
- Output the best match

After reading in the signal, the first step is to normalize the incoming signal around the x-axis and in amplitude. Normalizing around the x-axis occurs by finding the median value of the entire signal, which is the DC component of the signal. This value is then subtracted from the entire signal which results in moving the signal down around the x-axis. The normalization of the amplitude is achieved by subtracting the minimum value of the signal from the entire signal then dividing the signal by the maximum value of this signal. The next step is to eliminate silence by removing any parts of the signal whose amplitude value falls under a certain threshold. Then the wavelet transform is used to produce the signal's decomposition into five octaves.

Next, the classification stage begins. This can be done using a variety of methods, and we chose template matching since it is an easy, direct technique, good for showing our concept. The template signals were compared to the input signal. For this paper, one voice was used in the template for each word. To perform matching, we used correlation. The signals used in the template all followed the algorithm presented above.

4. RESULTS

Our experiment used two methods for speech recognition. First, we used correlation, to give us a simple method to compare against. Next, we used the DWT with correlation, and found that there was an improvement. The DWT naturally takes some warping into account, since it uses different scales of the input data. That is, a difference at one scale becomes half as large at the next.

Results were found for using the discrete wavelet transform on each of the five octaves produced. Thirty six phonemes were chosen as the template. Each phoneme was recorded five times and each of these was part of the test. There are a total of 175 possible correct matches for each experiment, five for each phoneme recorded. The input to the experiments is the phoneme that is to be recognized, while the output is a list of possible matches with the corresponding correlation values.

Figure 2 shows the number of correct matches for each octave as well as for the non-discrete wavelet transform experiments. "No DWT - 1st" represents the experiment which used the first group of samples of the input signal for matching, and "No DWT - 2nd" represents the experiment of using the middle group of samples. The reason for two different tests is that the first "No DWT" results were very poor, so we tried using the samples in the middle to see if this would improve performance. It did not.

As seen on the chart, use of the discrete wavelet transform has improved the number of correct matches from 4 to 30, at the very least. The greatest improvement is noticed in the first through third octaves using the approximation coefficients as the input to the wavelet transform. There is a marked decrease in the amount of correct matches as the number of octaves increases. There are 35 phonemes in our template, and the test data has 5 recordings for each phoneme. Therefore, the number of phoneme matches are out of a possible 175. The highest number of top matches was found to be from using the second octaves detail coefficients, with a 77 out of 175 or 57% correctness.

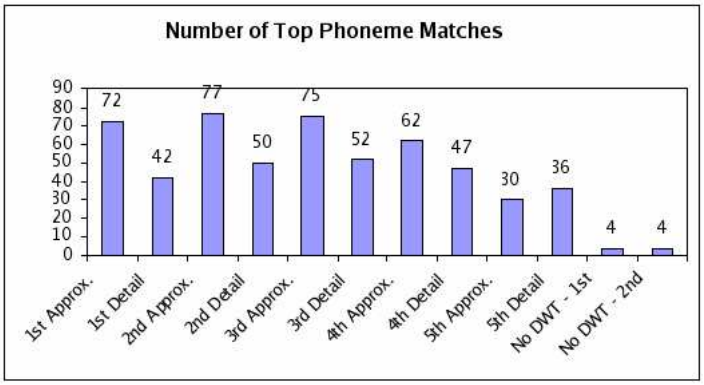

Fig. 2. Number of Correct Matches

Since our algorithm gives a value matching the input to each phonme, we also looked at whether the correct phoneme appeared in the top 5 or 10 matches. The greatest number of phoneme matches found in the top five was from the first octave approximation coefficients, which resulted in a correctness rate of about 123 out of 175 or 72%. For matches in the top 10, the highest number of matches is from using the third octave approximation coefficients. Not using the discrete wavelet transform in the feature extraction stage produced the lowest results. The highest number of phoneme matches in the top ten was from using the third octaves approximation coefficients, with a correctness rate of 150 out of 175 or about 86%.

From the results, phonemes which fall into the group of plosives were the most correctly matched. Examples of plosive phonemes are the 'b' sound in the word 'bad' and the 'p' sound in the word 'pad'. Other groups of phonemes which faired well overall were the clipped (e.g. 'a' as in apple), extended (e.g. 'ee' as in me), dipthong (e.g. 'i' as in eye), and syllabic (e.g. 'h' as in her). The unvoiced group (e.g. 'ch' as in cheese) had the least amount of matches throughout all of the experiments. One reason for different groups of phonemes matching better than others could be due to the phonemes lengths. All phonemes have a different length. Those closer in sound tend to have a length closer to each other. Normalizing the signals to all be around the same length may lose some data which would be missed for the classification stage.

Table 1 shows the groups of speech, and the number of correct matches for each octave using the approximation and detail coefficients as input to the classification stage. The numbers next to each of the phonemes represent how many times that particular phoneme was recognized out of five attempts.

The best all-around results seem to come from using the third octaves approximation coefficients for the classification stage. This observation is not only based on a high amount of correct matches but also from the fact that using this octave also results in less time for checking for a match. This is due to the original signal's coefficients being almost halved for each octave that is produced by the wavelet. Obviously, not using any method of feature extraction showed the poorest results of all with hardly any correct matches.

The highest percentage of correctness for the approximation coefficients occurs for the plosive and dipthong groups at 44.7% and 43% respectively. The lowest occurs for the unvoiced group at 24%. For using detail coefficients, the highest percentage of correctness is for the clipped phoneme group at 38.4%. The lowest percentage is for the plosive group at 16.7%.

The drop-off in correct matches after the third octave can be explained by the loss of too much frequency information. Because the wavelet transform smoothes out the frequency values, as more and more octaves are generated, the signals

	Octave 1 A, D	Octave 2 A, D	Octave 3 A, D	Octave 4 A, D	Octave 5 A, D
Clipped					
'a'	1, 2	1, 2	2, 3	0, 2	0, 0
'e'	3, 0	3, 0	3, 0	1, 2	0, 2
'i'	1, 3	1, 2	2, 2	1, 1	0, 1
'o'	3, 3	3, 3	3, 3	3, 3	2, 2
'uu'	3, 1	3, 4	3, 2	2, 2	0, 3
Extended					
'er'	0, 1	0, 0	1, 1	0, 2	0, 0
'ee'	2, 1	2, 0	1, 3	1, 1	0, 0
'uh'	4, 0	4, 2	4, 3	4, 2	0, 4
'oo'	3, 1	3, 1	4, 1	3, 3	2, 1
Dipthong					
'ai'	1, 4	1, 0	1, 0	0, 0	0, 0
'ay'	4, 1	4, 2	4, 3	4, 4	1, 1
'oe'	3, 1	2, 2	2, 2	3, 1	0, 0
'ow'	3, 0	3, 0	3, 2	3, 1	1, 2
Plosive					
'b'	2, 1	2, 1	2, 1	1, 1	0, 2
'p'	5, 1	5, 2	5, 0	5, 1	4, 0
'd'	2, 1	2, 3	2, 3	1, 1	0, 2
't'	0, 0	3, 0	4, 0	4, 0	4, 0
'k'	1, 0	1, 0	1, 0	1, 0	1, 0
'g'	2, 1	2, 1	2, 1	2, 1	1, 1
Voiced					
'th'	3, 3	3, 3	3, 2	3, 3	1, 3
'v'	1, 0	1, 0	1, 0	1, 3	0, 2
'j'	2, 1	2, 1	2, 2	2, 1	2, 0
'z'	1, 2	2, 5	3, 1	3, 3	2, 1
Unvoiced					
'f'	1, 1	1, 1	1, 1	1, 1	1, 1
'ch'	0, 0	0, 0	0, 0	0, 0	1, 0
'sh'	3, 1	3, 0	0, 3	0, 0	0, 0
's'	1, 2	3, 4	3, 2	3, 1	2, 1
Syllabic					
'h'	1, 1	1, 1	1, 1	1, 1	1, 1
'l'	0, 0	0, 2	0, 0	0, 0	0, 0
'm'	2, 1	1, 0	2, 3	3, 0	2, 1
'n'	0, 0	0, 0	0, 0	0, 0	0, 0
'ng'	0, 0	0, 0	0, 0	0, 0	0, 0
'w'	4, 0	4, 1	2, 0	0, 4	0, 1
'y'	4, 2	4, 3	2, 4	1, 0	0, 1
'r'	5, 5	5, 0	5, 1	4, 3	2, 2

Table 1. Correct matches for each octave using approximation (A) and detail (D) coefficients

generated by the transform are smoother.

5. CONCLUSIONS

The approach taken in this paper is to use the wavelet transform to extract coefficients from phonemes and to use cross-correlation to classify the phoneme. Cross-correlation measures the similarities between two signals. Normalization of the amplitudes and frequencies are used. Silence is eliminated from the signal as well. A Daubechies 8 wavelet is used to obtain five octaves of each signal. A template of each phoneme trying to be recognized is used in the matching process. The system is speaker dependent to make things simpler.

The results show that using the wavelet transform improved the accuracy in correctly identifying the phonemes over not using any method for feature extraction. The results also show that using the approximation coefficients to generate octaves in the wavelet transform give better accuracy than using the detail coefficients. The first three octaves give the best results, while the accuracy of using the fourth and fifth octaves declines.

The results demonstrate that it is possible to build a speech recognition engine using the wavelet transform and wavelet coefficients. Template matching was used for an easily designed way to compare and get results but would not be the best or most reliable method to use for building a recognition engine.

Future work includes first differentiating the type of phoneme such as voiced, unvoiced, or transitory and possibly taking a different action or evaluating a different octave of the signal depending on the type [11]. Since the approximation coefficients seem to give a better match in most cases, a score based on a combination of the approximation and detail coefficients should be investigated. For example, 75% of the score could be found from the approximation coefficients while the other 25% could be found from the detail coefficients. More work also needs to be done in signal normalization to create a speaker independent system.

6. REFERENCES

[1] Mukund Padmanabhan, and Michael Picheny, "Large-Vocabulary Speech Recognition Algorithms," *Computer*, April 2002, pages 42-50.

[2] Evandro B. Gouva, Pedro J. Moreno, Bhiksha Raj, Thomas M. Sullivan, and Richard M. Stern, "Adaptation and Compensation: Approaches to Microphone and Speaker Independence in Automatic Speech Recognition," *Proc. DARPA Speech Recognition Workshop*, February 1996, pages 87-92.

[3] Neal Leavitt, "Let's Hear It for Audio Mining," *Computer*, October 2002, pages 23-25.

[4] P.J. Jang and A. G. Hauptmann, "Learning to Recognize Speech by Watching Television," *IEEE Intelligent Systems*, Volume 14, No. 5, 1999, pp. 51-58.

[5] Amara Graps, "An Introduction to Wavelets," *IEEE Computational Science and Engineering*, Vol. 2, Num. 2, 1995.

[6] Selina Chu, Eamonn Keogh, David Hart, Michael Pazzani, "Iterative Deepening Dynamic Time Warping for Time Series," *Second SIAM International Conference on Data Mining*, Arlington, VA, April 11-13, 2002.

[7] C.J. Long, S. Datta, "Wavelet Based Feature Extraction for Phoneme Recognition," *Proceedings International Conference on Spoken Language Processing*, Volume 1, October 1996, pages 264-267.

[8] Gouva, E.B., Stern, R.M., "Speaker Normalization Through Formant-Based Warping of the Frequency Scale", *5th European Conference on Speech Communication and Technology*, Volume 3, September 1997, pages 1139-1142.

[9] Bhiksha Raj, Evandro B. Gouvea, and Richard M. Stern, "Cepstral Compensation by Polynomial Approximation for Environment-Independent Speech Recognition," *Proc. International Conference on Spoken Language Processing*, 1996, pages 2340-2343.

[10] A. Ganapathiraju, J. Hamaker, M. Ordowski, G. Doddington and J. Picone, "Syllable-Based Large Vocabulary Continuous Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 4, May 2001, pp. 358-366.

[11] E. Shlomot, V. Cuperman, and A. Gersho, "Hybrid Coding of Speech at 4 Kbps", *Proceedings 1997 IEEE Workshop on Speech Coding*, Pocono Manor, PA, September 1997, pahes 37-38.