Phoneme segmentation of speech

Bartosz Ziółko, Suresh Manandhar and Richard C. Wilson
Department of Computer Science, University of York, UK
bziolko,suresh,wilson@cs.york.ac.uk

Abstract

In most approaches to speech recognition, the speech signals are segmented using constant-time segmentation, for example into 25 ms blocks. Constant segmentation risks losing information about the phonemes. Different sounds may be merged into single blocks and individual phonemes lost completely.

A more satisfactory approach is to attempt to segment the phoneme boundaries from the speech signals and use these boundaries to define blocks. The discrete wavelet transform (DWT) is interesting in the analysis of speech since it is easy to extract parameters which take into account the properties of the human hearing system. The analysis of the power in different frequency bands offers potential for distinguishing the start and end of phonemes. For many boundaries, there is no discernible drop in overall power, and at some frequencies, the power is broadly constant over the lifetime of the phoneme. However, many phonemes exhibit rapid changes in particular subbands which can be used to detect their start and endpoints.

In this paper we apply the DWT to speech signals and analyse the resulting power spectrum and its derivatives to locate candidates for the boundaries of phonemes in continuous speech. We compare the results with hand segmentation and constant segmentation over a number of words. The method proves effective for finding most phoneme boundaries.

1 Introduction

In most approaches to speech recognition, the speech signals need to be segmented before recognition can take place. The properties of the signal contained in each segment is then assumed to be constant, or in other words to be characteristic of a single part of speech.

The most often used current method is to use constant-time segmentation, for example into 25 ms blocks [8]. This methods benefits from simplicity of implementation and the ease of comparing blocks of the same length. Clearly, however, the boundaries of speech elements such as phonemes do not lie on fixed position boundaries; phonemes naturally vary in length both because of their structure and due to speaker variations. Constant segmentation therefore risks losing information about the phonemes. Different sounds may be merged into single blocks and individual phonemes lost completely.

A more satisfactory approach is to attempt to segment the phoneme boundaries from the speech signals and use these boundaries to define blocks. A number of approaches have previously been suggested for this task [2, 7, 9] but these utilise features derived from acoustic knowledge of the phonemes. Such methods need to be optimised to particular phoneme data and the performance is often not as good on new speech data. Other pattern recognition approaches such as neural nets [5] have also been tried, but these also require training. Another approach for segmentation is by applying Segment Models instead of HMM [3]. This solution group frames into segment-like sequences of frames using modelling.

Spectral analysis of the speech signal is the most appropriate method for extracting information from speech signals. DWT has been successfully used in many signal processing applications including speech [1, 4, 6] for the spectral analysis of data. The analysis of the power in different frequency bands offers potential for distinguishing the start and end of phonemes. For many boundaries, there is no discernible drop in overall power, and at some frequencies, the power is broadly constant over the lifetime of the phoneme. However, many phonemes exhibit rapid changes in particular subbands which can be used to detect their start and endpoints.

The outline of this paper is as follows; in section 2 we describe the discrete wavelet transform and the derivation of power envelopes from the DWT. In section 3 we discuss the segmentation of phoneme start and end points from the power curves. Finally, in section 4 we evaluate the effectiveness of the proposed method and illustrate its advantages over constant segmentation.

2 The Discrete Wavelet Transform

DWT is interesting in the analysis of speech since it is easy to extract parameters which take into account the properties of the human hearing system [6]. The discrete wavelet

transform is computed in the normal way from the coefficients of the basis expansion:

$$s(t) = \sum_i c_{m+1,i}\psi_{m+1,i}(t) \quad (1)$$

where $\psi_{m+1,i}$ is the ith orthogonal basis function at the $(m+1)$th resolution level.

The coefficients of lower levels may be calculated from the well known formulae [4]

$$c_{m,n} = \sum_i h_{i-2n}c_{m+1,i} \quad (2)$$

$$d_{m,n} = \sum_i g_{i-2n}c_{m+1,i} \quad (3)$$

which allow efficient computation of the transform at different scale levels. h and g are derived from the appropriate scale function ψ. The elements of the DWT for a particular level may be collected into a vector, for example $\mathbf{d}_m = (d_{m,1}, d_{m,2}, ...)^T$. The formulae (2) and (3) decompose the signal into lower resolution and frequency levels. Once the coefficients $c_{m+1,i}$ are computed for the $(m+1)$th resolution level, we can iteratively apply (2) and (3) to obtain the subsequent coefficients.

By using a filter bank, the wavelet spectra are produced by cascading the filtering and downsampling operations in a tree-structure. The root of the tree consists of the coefficients of the wavelet expansion of the original speech signal. Subsequent levels in the tree are constructed by recursively applying the wavelet transform to split the signal into an approximation and detail part. Ultimately, we need only keep one approximation at the lowest level and the remaining detail parts in order to describe the signal. The DWT is therefore represented by a set

$$\text{DWT}(s) = \{\mathbf{d}_m, \mathbf{d}_{m-1}, ..., \mathbf{d}_{m-M-1}, \mathbf{c}_{m-M-1}\} \quad (4)$$

The discrete Meyer wavelet was chosen as the basis for the DWT because of symmetry and compact support in the frequency domain.

The number of samples at each level of the DWT varies; Table 1 details the number of samples at each level relative to the lowest resolution level. In order to calculate the signal power and provide a representation with the same time interval between samples, we compute the power over a varying number of samples relative to each resolution level:

$$p_m(i) = \sum_{j=\alpha i}^{\alpha(i+1)-1} d_{m,j}^2 \quad (5)$$

α is number of samples for every sample on lowest DWT level.

The power derived from the DWT sub-bands will still show rapid variations; in order to obtain an overall power we find the envelope of the power by choosing the maximum value within a sliding window. The windowing scale varies with the scale of the DWT and is detailed in Table 1.

3 Phoneme detection

Many phonemes are represented by a rapid rise and fall in power of one or more frequency sub-bands. For this reason we turn to the first derivative of power. The first order differences in the power are inevitably noisy, and so we use a smoothed differencing operator. The sub-band power is convolved with the mask $[1, 2, -2, -1]$ to obtain smoothed rate-of-change information.

Clearly, we would expect the rate-of-change of power to be large at the beginning and end of phonemes, in at least some of the sub-bands. However, this does not uniquely define start and end points, for two reasons. Firstly, the power can rise over a considerable length of time at the start of a phoneme, leading to an ambiguous start time. Secondly, there may also be rapid changes in power in the middle of a segment. A better method of detecting the boundary of phonemes relies on power transitions in the DWT transform. The start of a phoneme should be marked by an initially small but rapidly rising power level in one or more of the DWT levels. In other words, we should expect the power to be small and the derivative to be large. We can detect these points by looking for crossings of the form

$$p_m(t) = \beta \left| \frac{dp_m(t)}{dt} \right| \quad (6)$$

where the constant β accounts for the time scale and sensitivity of the crossing points. For practical reasons in our method we seek for indexes for which the smoothed power and derivative approach close to each other instead of crossing.

The boundaries produced from the derivative do not precisely define the start or end of a phoneme for a number of reasons. Firstly, we must analyse all frequency bands, and the phoneme boundaries may not be apparent in all bands. In fact, for some phonemes, only one frequency band may show significant variations in power. Furthermore, the precise positions of the boundaries may vary slightly between levels. Secondly, despite smoothing the derivative, near the threshold there may be a number of transitions which represent the beginning or end of the same phoneme. These must be grouped together. These problems are addressed by grouping together all transition points across all the bands, provided they are less than time α apart where α represents the minimum length of a phoneme. Typically this is 5 which stands for 29 ms. The boundary position is the average of these grouped transition points.

The algorithm consists of following steps:

1. Normalise a speech signal.
2. Decompose a signal into six levels of the DWT.

Table 1. Characteristics of the discrete wavelet transform levels

DWT Level	Frequency band (Hz)	Samples (per sample at level 6)	Window size ω
1	2756–5512	32	3
2	1378–2756	16	3
3	689–1378	8	3
4	345–689	4	5
5	172–345	2	7
6	86–172	1	9

3. Calculate the sum of power samples in all frequency subbands according to Table 1 to obtain the power representations $p_m(n)$ of the mth subband (5).
4. Calculate the envelopes p'_m for power fluctuations in each subband by choosing the highest values of p_m in a window of given size ω (Fig. 1 and Table 1).
5. Calculate the first derivative $d_m(i)$ of $p'_m(i)$.
6. Given a threshold p, find indexes for which $||d_m(i)| - p'_m(i)| < p$ *AND* $(||d_m(i+1)| - p'_m(i+1)| > p$ *OR* $||d_m(i-1)| - p'_m(i-1)| > p)$. Write such indexes in one vector (marked as asterisks in Fig. 1).
7. Find and group indexes where there is no space between neighbouring ones longer than attribute α.
8. Calculate an average index value for each group found in the previous step as the representative of a group. They are indexes of phonemes' boundaries in indexing order of DWT level 6. For the original signal indexes have to be multiplied by 32.

4 Experimental results

In order to assess the quality of our results, we have hand-segmented 43 Polish words for comparison. The hand segmentation itself is not an entirely accurate process, since there may be a degree of uncertainty precisely where the phoneme starts and ends, to within a few samples. The words are also segmented using our automatic technique and a constant segmentation method where the speech is broken into fixed length segments.

The quality of segmentation may be assessed on two criteria. Firstly, the right number of segments should be found - the number of segments should correspond to the number of phonemes present in the speech. The error in the number of segments for word w is defined to be

$$\epsilon_n(w) = \frac{|n_a - n_h|}{n_h} \tag{7}$$

where n_a and n_h are the number of segments in the automatic and hand segmentation respectively.

The second criterion is accuracy of the position of the segmentation. This is based on the closeness of the boundary to the hand-segmented boundary. Since we do not know which boundary corresponds to a particular boundary in the hand segmentation, we take the closest boundary as the correct one. The error in placement for word w is

$$\epsilon_p(w) = \sum_j \min_i |p_j - q_i| \tag{8}$$

where p_i is the position of the ith boundary in the automatic segmentation, and q_j is the jth boundary position in the hand segmentation. Finally, we construct and overall error of

$$\epsilon(w) = \frac{1}{n_w} \sum_w 5\epsilon_n(w) + \epsilon_p(w) \tag{9}$$

where n_w is the number of words in evaluation set (43 in our example). The overall error for our dataset is summarised in Table 2.

Table 2. Comparison on constant segmentation and proposed method

Method	ϵ_n	ϵ_p	Overall error
Const 23.2 ms	125.5	243	20.2
Const 92 ms	3.6	227	5.7
proposed method	6.4	152	4.3

Table 3. Results for one word

Method	Segment positions							
Auto	0	4	38	54		86	103	118
Hand	0	4	27	52	66	86	105	117
	$\epsilon_n = 0.125$ $\epsilon_p = 2.29$							

Fig. 1 shows and example of the segmentation process. The six wavelet bands are shown, along with the automatic and hand segmented boundaries. For this word, the boundary positions are shown in Table 3. The method finds nearly all the boundaries accurately, to within 2 samples, but fails to find one boundary and misplaces one boundary too far to the end of one phoneme.

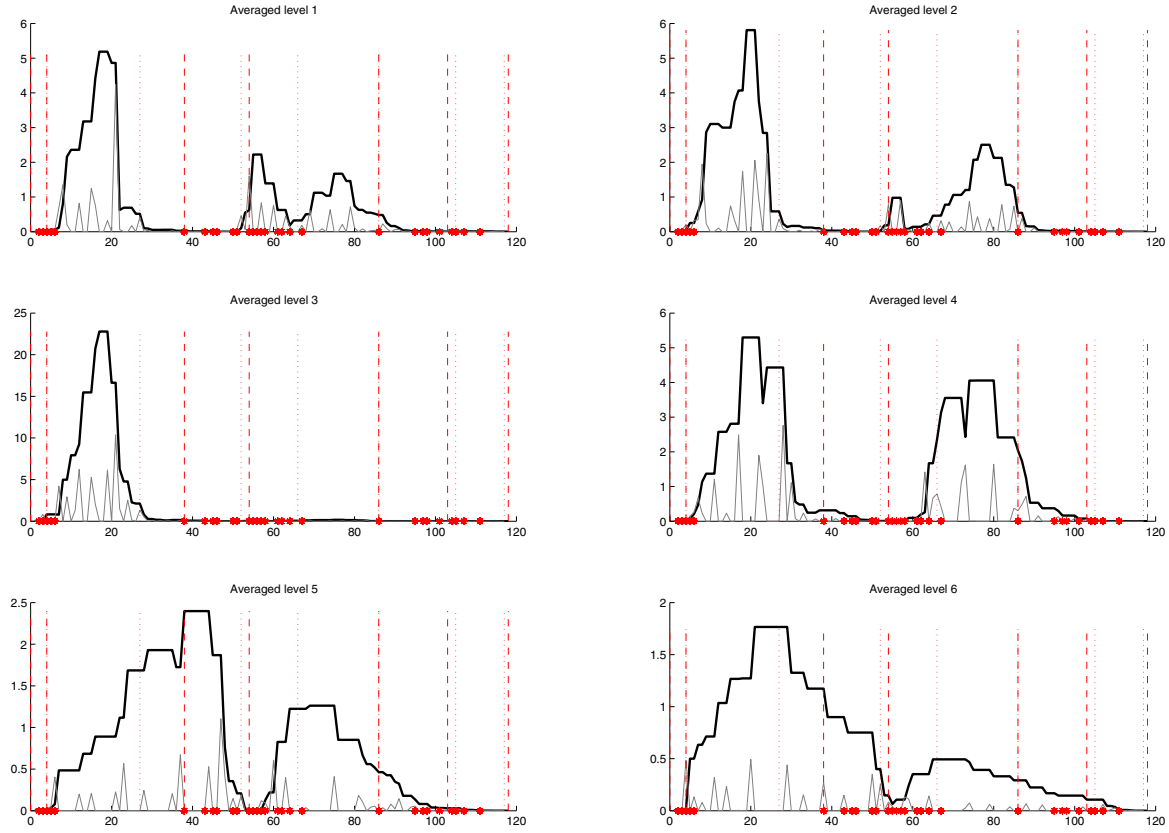

Figure 1. An example of the segmentation of a name 'Andrzej' /ɑːn dʒeɪ/. 6 DWT levels are presented. Dotted lines are hand segmentation boundaries; dashed lines are automatic segmentation boundaries, bold lines are power envelopes and thin grey lines are absolutes of power envelope first derivative. Asterisks are indexes with fulfiled condition for boundary candidate (see step 6 of the algorithm).

References

[1] O. Farooq and S. Datta. Wavelet based robust subband features for phoneme recognition. *IEE Proceedings: Vision, Image and Signal Processing*, 151(3):187–193, 2004.

[2] D. B. Grayden and M. S. Scordilis. Phonemic segmentation of fluent speech. *Proc. of ICASSP*, pages 73–76, 1994.

[3] M. Ostendorf, V.V. Digalakis, and O.A. Kimball. From hmm's to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 4:360–378, 1996.

[4] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE Signal Processing Magazine*, 8:11–38, 1991.

[5] Y. Suh and Y. Lee. Phoneme segmentation of continuous speech using multi-layer perceptron. In *ICSLP 96*, 1996.

[6] D. Wang and S. Narayanan. Piecewise linear stylization of pitch via wavelet analysis. *Proc. of Interspeech*, 2005.

[7] C. J. Weinstein, S. S. McCandless, L. F. Mondshein, and V. W. Zue. A system for acoustic-phonetic analysis of contimuous speech. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 23:54–67, 1975.

[8] S. Young. Large vocabulary continuous speech recognition: a review. *IEEE Signal Processing Magazine*, 13(5):45–57, 1996.

[9] V. W. Zue. The use of speech knowledge in automatic speech recognition. *Proc. of the IEEE*, 73:1602–1615, 1985.