Vocal command signal segmentation and phonemes classification.

E. Wesfreid [a,*] V. Wickerhauser [b,**]

[a] *LMPA, Université du Littoral. Calais - France.*
E-mail: eva@lma.univ-littoral.fr
[b] *Department of Mathematics, Washington University, St. Louis - USA.*
E-mail: victor@math.wustl.edu

In this paper we study a set of *vocal command signals* recorded in a noisy environment. We describe and use Fang's segmentation algorithm[6] to isolate *near phonemes*. A *piecewise* constant time-frequency spectrum for the *near phonems* is then computed using the smooth *cosine4* orthonormal basis[3,7,8,13] defined over the segmented time axis. This paper proposes a criterion to distinguish phonemes using this *smooth local spectrum*.

Keywords: signal processing, wavelets, time-frequency representation.

AMS Subject classification: 94A11, 94A12

1. Introduction

A signal can be decomposed into a linear combination of elementary *waveforms*, where each waveform is essentially supported by a rectangle $R = [a, b] \times [\alpha, \beta]$ in the time-frequency plane. One now has available a large selection of waveforms or *time-frequency atoms* (for example, windowed Fourier functions, Malvar–Wilson bases[3,7,8], and wavelet packets[9,10]). Since the choice of *time-frequency atoms* is not unique, the decomposition can be adapted to the analysed signal.

Speech signals may be regarded as a sequence of overlapping phonemes, and one goal of time-frequency representation is to isolate and analyze these phonemes. The windowed Fourier representation has played a major role in speech processing[1,12], and more recently, other representations based on wavelets have been used with varying success, depending on the objective of the analysis.

The local trigonometric *Best Basis* of Coifman and Wickerhauser[4] is a fast algorithm that computes a local spectrum based on a dyadic segmentation of the time axis in $O(NlogN)$ operations, where N is the number of sample points. We used this algorithm in a previous paper[15] for speech signal analysis and segmentation and it proved to be successful for speech compression. It is not, however, well suited for isolating phonemes, since there is no reason for phonemes to "begin" and "end" at dyadic points.

In this paper we use Fang's segmentation algorithm to segment the time axis. The purpose of this algorithm is to isolate the phonemes. The algorithm is based on the measurement of an *instantaneous frequency*, and it places segmentation points where it detects a change in this *instantaneous frequency*. Thus it tends to isolate phonemes. Since at best we can only expect these point to be approximate, we say that the segments contain *near phonemes*.

We use a local trigonometric basis to compute the (piecewise constant) spectra of these *near phonemes*. The algorithm takes $O(N^2)$ operations, and this analysis is useful for synthesis, compression, and classification.

* The authors wish to thank Robert Ryan for helpful discussions and suggestions.
** Research supported in part by AFOSR, NSF, and the Southwestern Bell Telephone Company.

2. Fang's segmentation algorithm

A segmentation of a sampled signal is a strictly increasing sequence of integers, which are the initial indices of each segment. Fang's segmentation algorithm computes the *local maxima* of a *frequency change function*; this function is the average of an *instantaneous frequency change function*.

2.1. Instantaneous frequency change function

This function can be obtained using the spectrum computed with either the *block* or the *smooth dct4* transform. In this paper we use the *smooth dct4* transform.
This function is the difference between the *flatness* of the spectrum over $[j-\ell, j+\ell]$ with $(\ell > 0)$ and the *flatness* of the combined spectra over $[j-\ell, j]$ and $[j, j+\ell]$. This *flatness* can be measured with one of the following *cost functions*

$$\lambda(x_0, x_1, \ldots, x_n) = \sum_{k=0}^{n-1} |x_k| \qquad (2.1)$$

or

$$\lambda(x_0, x_1, \ldots, x_n) = -\sum_{k=0}^{n-1} |x_k|^2 log(|x_k|^2). \qquad (2.2)$$

Let A_j, B_j, and C_j denote the *dct4* spectrum over $[j-\ell, j]$, $[j, j+\ell]$ and $[j-\ell, j+\ell]$. Then

$$IFJ(j) = \lambda(C_j) - (\lambda(A_j) + \lambda(B_j)), \qquad (2.3)$$

where $j \in \{\eta + \ell, \ldots, N - \eta - \ell\}$, is the *instantaneous frequency change* function. This function oscillates even when the signal is periodic, as shown in Fig.1. The IFC function is shown in the bottom. The filtered version of this function is plotted in the middle.

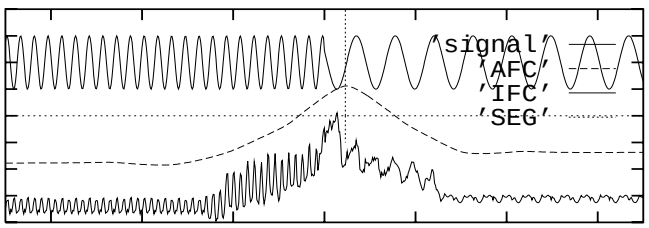

Figure 1. IFC and AFC frequency change functions

2.2. Segmentation algorithm

This algorithm consists of the following five steps:

1. Compute $IFC(j)$ for $j \in]\ell + \eta, N - \ell - \eta[= I$ as follows:

   Consider $IFC(j) = 0 \ \forall j \in I$ and compute C_j, the *dct4* transform of the signal over $[j-\ell, j+\ell]$, and B_j, the *dct4* transform of signal over $[j, j+\ell]$. Then

   $$IFC(j) = IFC(j) + \lambda(C_j) - \lambda(B_j),$$

   and

   $$IFC(j+\ell) = IFC(j+\ell) - \lambda(B_j),$$

   since $A_{j+\ell} = B_j$.

2. Filter $IFC(j)_{j \in I}$ to obtain an *averaged frequency change function* $AFC(j)_{j \in I}$ as follows:

   If H and G denote a *biorthogonal lowpass filter* and its dual, then

   $$AFC = G^d H^d (IFC),$$

   where $H^d = HHH \ldots H$.

3. Find the *local maxima* by detecting zero crossings of the adjacent differences of $AFC(j)_{j \in I}$.

4. Squelch the local maxima above some threshold.

5. **Improvement**
   We consider only the local maxima of AFC such that its second derivative is lower than a given negative threshold. This condition eliminates those maxima that are too flat.

There are three parameters to set:
1) the adjacent window overlap η,
2) the window size ℓ,
3) the number d of iterations of the *lowpass filter* H.

In particular, we use this algorithm with $\eta = 16$, $\ell = 256$, and $d = 9$ to obtain a *near phoneme* segmentation of *noisy vocal signals* recorded in flight.

A *vocal command signals* recorded in a noisy environment was segmented using this algorithm. Fig. 2 and Fig. 3 show this signal at the top, the IFC function at the bottom, the AFC function in the middle and the segmentation with vertical lines at the AFC local maxima:

$$0 = a_0 < a_1 < \ldots < a_s = N.$$

This is a *non dyadic* segmentation that tends to isolate *near phonemes*.

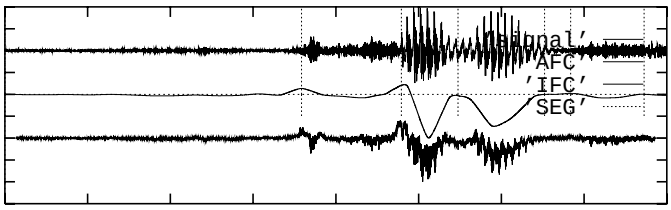

Figure 2. vocal signal segmentation

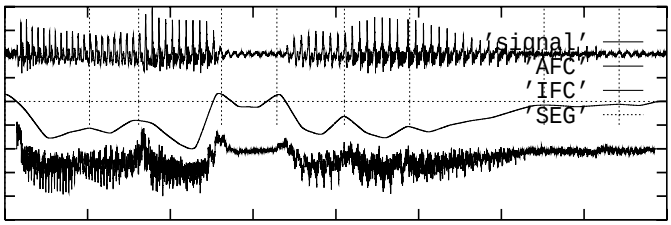

Figure 3. vocal signal segmentation

3. Block and smooth dct4 transform

The *block dct4 spectrum* of a signal S over a segmented time axis,

$$0 = a_0 < a_1 < \ldots < a_s = N,$$

is the set of coefficients

$$D_j = \{d_{j,k} : 0 \leq k < \ell_j\} \tag{3.1}$$

in the decomposition

$$S(t) = \sum_{\substack{j \in Z \\ 0 \leq k < N}} d_{j,k}\phi_{j,k}(t),$$

where

$$d_{j,k} = \langle S, \chi_{I_j}\phi_{j,k}\rangle$$

is the *block dct4* transform,

$$\phi_{j,k} = \frac{\sqrt{2}}{\sqrt{|\ell_j|}} \cos\frac{\pi}{|\ell_j|}(k+\frac{1}{2})(t-a_j)$$

is the *cosine4* function, and $\chi_{I_j}(t)$ is the indicator function of I_j, which is 1 in $I_j = [a_j, a_{j+1}]$ and 0 outside.

We are going to describe the *smooth dct4* transform algorithm that computes the *smooth local spectrum* of a sampled signal $\{f(j)\}_{0 \leq j < N}$ over a segmented time axis:

$$0 = a_0 < a_1 < \ldots < a_s = N.$$

We consider the following functions and sets:

- the *raising function*

$$r(t) = \begin{cases} 0 & t \in]-\infty, -1[\\ \sin[\frac{\pi}{4}(1+\sin\frac{\pi}{2}t)] & t \in [-1, 1] \\ 1 & t \in [1, \infty[\end{cases}$$

- the following *orthogonal* window

$$\omega_j(t) = r(\frac{(t-a_j)}{\eta})r(\frac{(a_{j+1}-t)}{\eta}) \tag{3.2}$$

over $I_j = [a_j, a_{j+1}]$, where $t \in Z + 1/2$, $0 < \eta < \ell_j/2$, and $\ell_j = (a_{j+1} - a_j)$ (η is the adjacent window overlap).

- $b_j(t) = \frac{r(t-a_j)}{\eta}$.
- $O_j^+ =]a_j, a_j + \eta[$, $O_j^- =]a_j - \eta, a_j[$.

We use the *folding*[14] operator

$$U_j f(t) = \begin{cases} b_j(t)f(t) + b_j(2a_j - t)f(2a_j - t) & if\ t \in O_j^+, \\ b_j(2a_j - t)f(t) - b_j(t)f(2a_j - t) & if\ t \in O_j^-. \end{cases}$$

and its adjoint, the *unfolding*[14] operator:

$$U_j^* f(t) = \begin{cases} b_j(t)f(t) - b_j(2a_j - t)f(2a_j - t) & if\ t \in O_j^+, \\ b_j(2a_j - t)f(t) + b_j(t)f(2a_j - t) & if\ t \in O_j^-. \end{cases}$$

that verify $U_j U_j^* = U_j^* U_j = id$, to compute the *folded* function

$$F_{a_j, a_{j+1}} = \chi_{I_j} U_j U_{j+1} f$$

and to compute the given *orthogonal* window (3.2). This window is equal to the rectangular window χ_{I_j} *unfolded* at a_j and at a_{j+1}:

$$\omega_j(t) = U_j^* U_{j+1}^* \chi_{I_j}. \tag{3.3}$$

Otherwise, given a time axis segmentation

$$Z = \bigcup_{j \in Z} I_j$$

with $I_j = [a_j, a_{j+1})$, such that $\inf_{j \in Z}(a_{j+1} - a_j) > 0$ the associated *orthonormal trigonometric basis*

$$\{\Psi_{j,k}(t)\}_{j \in Z, 0 \leq k < \ell_j},$$

where

$$\Psi_{j,k}(t) = w_j(t)g_{j,k}(t) \tag{3.4}$$

consists of *orthogonal* windows $w_j(t)$ modulated by the *cosine4* functions as is shown in Fig. 4. The *smooth spectrum* of f over $I = [a_j, a_{j+1}]$ is the set of coefficients

$$C_j = \{c_{j,k} : 0 \leq k < \ell_j\}$$

of the signal decomposition:

$$f(t) = \sum_{\substack{j \in Z \\ k \in N}} c_{j,k}\Psi_{j,k}(t),$$

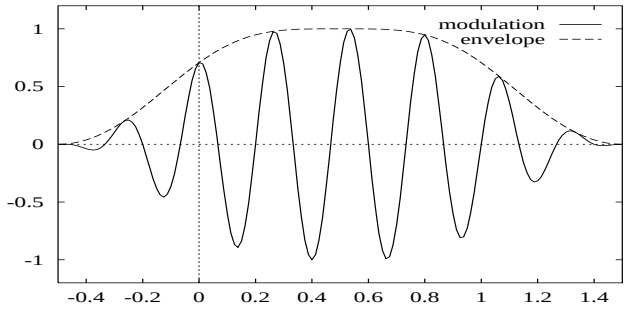

Figure 4. Lapped orthogonal basis function

where

$$c_{j,k} = \langle f, \Psi_{j,k} \rangle = \langle f, \omega_j g_{j,k} \rangle \tag{3.5}$$

is the *smooth dct4 transform.*
Since

$$c_{j,k} = \langle f, U_j^* U_{j+1}^* \chi_{I_J} g_{j,k} \rangle = \langle F_{a_j, a_{j+1}}, g_{j,k} \rangle,$$

the *smooth dct4* transform $c_{j,k} = \langle f, \Psi_{j,k} \rangle$ is equal to the *block dct4* transform of the folded signal:

$$c_{j,k} = \langle F_{a_j, a_{j+1}}, g_{j,k} \rangle. \tag{3.6}$$

4. Smooth spectrum near phonemes

In a previous paper[15], speech signals were analyzed using the orthonormal trigonometric *Best Basis* of Coifman and Wickerhauser, based on a *split and merg*[10] algorithm. Since phoneme length is not necessarily dyadic, we have used Fang's segmentation algorithm and then compute the *smooth dct4* transform over the segmented signal. Fig. 5 and Fig. 6 show the *smooth local spectrum*, in absolute value, over the segmented signal. The coefficients of this spectrum are defined in (3.5) and (3.6) over each interval $[a_j, a_{j+1}]$.

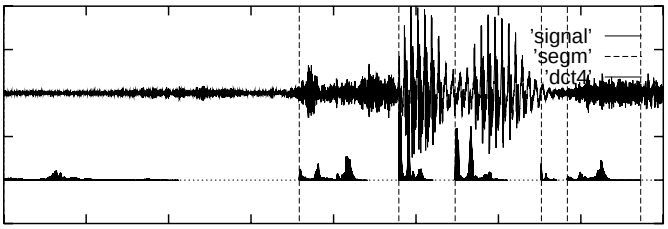

Figure 5. Smooth local spectrum near phonemes

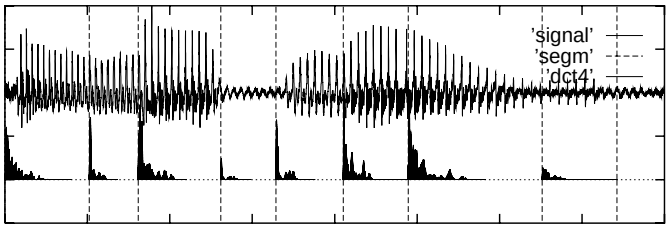

Figure 6. Smooth local spectrum near phonemes

5. Classification

In this paper we have studied a set of *vocal command signals* recorded in a noisy environment. We first segment each signal into *near phonemes* and then compute either the *block* or the *smooth local spectrum.*
We use the *frequency center of mass* to distinguish some unvoiced phonemes like /s/ from voiced phonemes like vowels.

We use the *spectrum envelope* to compare[2] voiced segments. This *spectrum envelope* is a cubic interpolation function defined over a set of local maxima extracted from the computed spectrum. We consider a

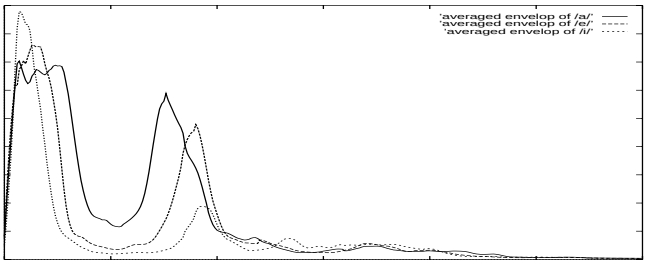

Figure 7. Averaged envelope

training set for each vowel and we compute an *averaged spectrum envelope* for each one.
We use this *averaged spectrum envelope*, to distinguish voiced phonemes. For instance, to recognize a given vowel among /a/, /e/, and /i/, we compare the *spectrum envelope* f of the given vowel with each *averaged spectrum envelope* g_x for $x \in \{a, e, i\}$ using the following function denoted by *cos*:

$$cos(f, g_x) = \frac{\sum_n f[n] g_x[n]}{\sqrt{\sum_n f^2[n] \sum_n g_x^2[n]}}.$$

We compute $cos(f, g_x)$ for $x \in \{a, e, i\}$ and we consider that the given vowel is *near* the *training* set of /a/ when $cos(f, g_a) > cos(f, g_x)$ for $x \in \{e, i\}$.
Phonemes /a/, /e/ and /i/ are compared in Fig. 8; the axis show the values of $cos(f, g_a)$, $cos(f, g_e)$ and $cos(f, g_i)$.

We used Fang's segmentation algorithm combined with the *smooth dct4* transform successfully to classify several phonemes.

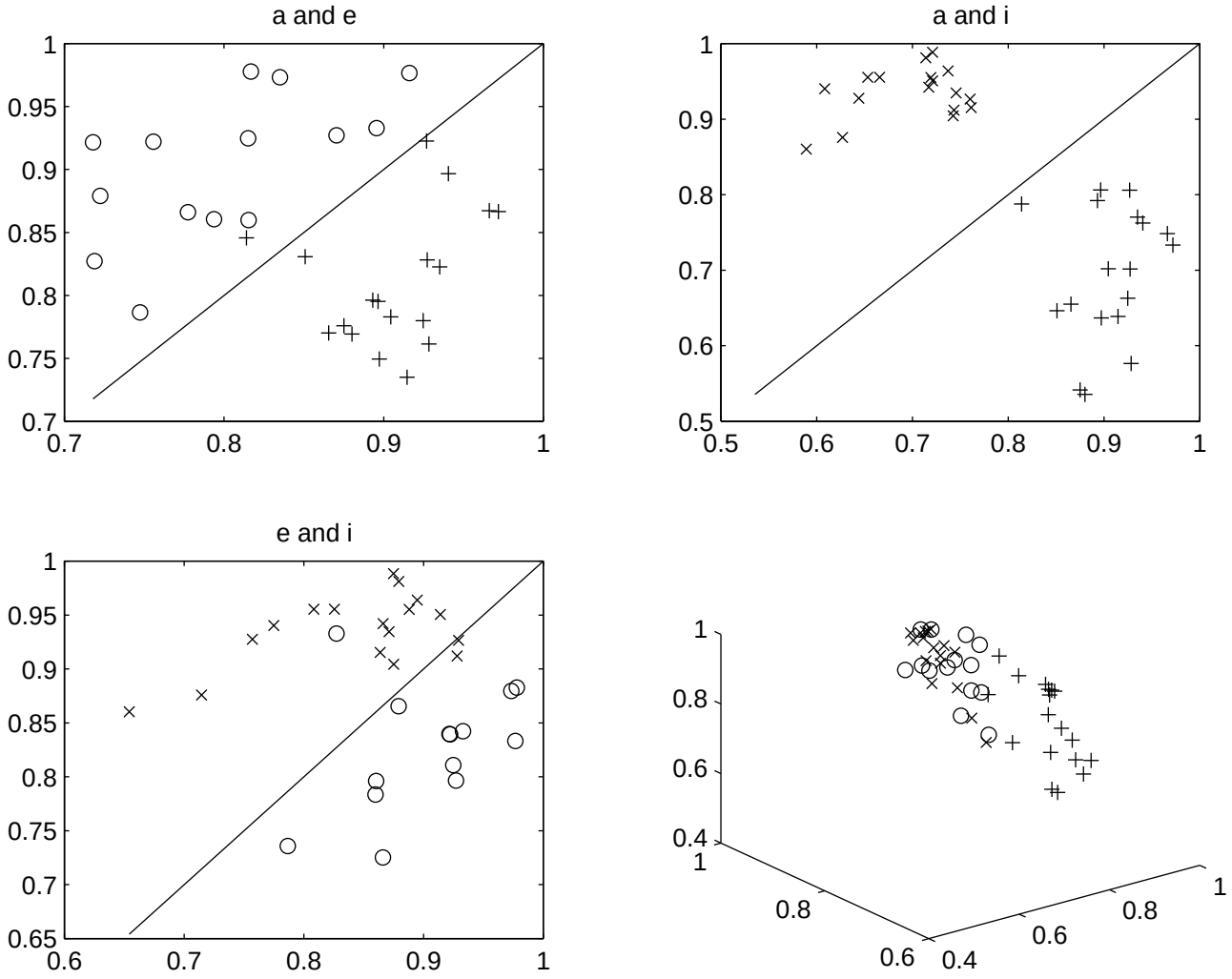

Figure 8. Phoneme classification

References

[1] Calliope, *La parole et son traitement automatique*, CNET, ENST, France, 1989.

[2] V. Camion, DEA report, Ecole Polytechnique, 1998.

[3] R.R. Coifman and Y. Meyer, *Remarques sur l'analyse de Fourier à fenêtre*, C. R. Acad. Sci. Paris 312, pp. 259-261, 1991.

[4] R.R. Coifman and M.V. Wickerhauser, *Entropy-based algorithms for best-basis selection*, IEEE Trans. Info. Theory, March, 1992.

[5] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.

[6] X. Fang, *Automatic Phoneme Segmentation of Continuous Speech Signals*, IEEE Transactions, 1994.

[7] H. Malvar, *Lapped transforms for efficient transform/subband coding*, IEEE Trans. Acoustics, Speech and Signal Processing, 38, pp. 969-978, 1990.

[8] H. Malvar, *Signal Processing with Lapped transforms*, Artech House, Norwood, MA, 1992.

[9] S. Mallat, *a Wavelet tour of signal processing*, Academic Press, 1998.

[10] Y. Meyer, *Wavelets: Algorithms and Applications*, Siam, 1993. Translated and Revised by R.D. Ryan.

[11] Y. Meyer, *Ondelettes et Algorithmes Conccurrents*, Hermann, 1992.

[12] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, 1978.

[13] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Prentice-Hall, 1995.

[14] V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, A.K. Peters, Wellesley, MA, 1994.

[15] E. Wesfreid and V. Wickerhauser, *Adapted trigonometric transform and speech processing*, IEEE Trans. Acoustic Speech Processing, Dec. 1993.