



A new technique to construct a wavelet transform matching a specified signal with applications to digital, real time, spike, and overlap pattern recognition [☆]

Rodrigo Capobianco Guido ^{*}, Jan Frans Willem Slaets, Roland Köberle, Lírio Onofre Batista Almeida, José Carlos Pereira

Department of Physics and Informatics, Institute of Physics at São Carlos and Department of Electrical Engineering, School of Engineering at São Carlos, University of São Paulo, Avenida Trabalhador São Carlense 400, 13560-970 São Carlos, SP, Brazil

Available online 10 May 2005

Abstract

This work describes a new and different path to create a wavelet transform that can match a specified discrete-time signal. Called *Spikelet*, it is designed and optimized to spike and overlap pattern recognition in the digitalized signal that comes from H1, a motion-sensitive neuron of the fly's visual system. The technique proposed here and the associated algorithm, implemented in real time using a digital signal processor (DSP), are fully detailed. The results obtained matching the signal under analysis show an improvement over all other transforms, including the Daubechies transform. This reassures the efficacy of our transform.

© 2005 Elsevier Inc. All rights reserved.

Keywords: Matched wavelet transform; Real time DSP; Pattern recognition; Time–frequency–shape analysis

[☆] Based on Rodrigo Capobianco Guido's Ph.D. thesis.

^{*} Corresponding author. Fax: +55 16 33739777.

E-mail addresses: guido@ifsc.usp.br (R.C. Guido), jan@ifsc.usp.br (J.F.W. Slaets), rk@ifsc.usp.br (R. Köberle), lirio@ifsc.usp.br (L.O.B. Almeida), pereira@sel.eesc.usp.br (J.C. Pereira).

URLs: <http://www.if.sc.usp.br>, <http://www.sel.eesc.usp.br>.

1. Introduction

1.1. Article organization

This article is organized as follows: Sections 1.2–1.5 describe the issue to be faced, a basic review on techniques for spike and overlap sorting, techniques of time–frequency analysis, discrete wavelet transforms, and a brief comment on two methods to construct wavelets matching a specified signal. Section 2 contains the characterization of the signal to match, called the *default spike*. Section 3 details the construction of the proposed wavelet, *Spikelet*. Section 4 describes the algorithm where Spikelet is used to spike and overlap sorting. Section 5 lists the results and conclusions.

1.2. Problem description

Much literature has appeared recently describing different techniques to create wavelet transforms that match a specified signal, for instance [1–5]. The technique proposed in this work was inspired by the Daubechies transform [6] together with inputs drawn from [7] and uses the least-squares method [8] to get the filter coefficients.

In the application treated here, the tests using Spikelet with support 4 showed that it was possible to distinguish at least 7 different patterns against no more than 5 patterns when using other wavelet transforms with support 20 (5 times larger). Furthermore, we determine in which portion of the signal the patterns occur, making a time–frequency–shape analysis.

The signals under analysis correspond to the neural action potentials of H1, a motion-sensitive neuron in the fly’s visual system [9]. They were acquired extracellularly, using a platinum–iridium electrode while the fly views a pattern of vertical random bars moving horizontally across its visual field. The experimental setup is shown in Figs. 1–4. Particularly, Fig. 1 shows the full experiment with the fly, the electrode, and the accessories. Figure 2 shows the Diptera laboratory where the experiment was developed; Fig. 3 shows a spike captured in real time on the oscilloscope. Figure 4 shows the pattern of bars acting as a stimulus to the fly’s visual system during the experiment.

Figure 5 shows the seven patterns easily classified by Spikelet. They are called, from left to right, *stressed left overlap*, *left overlap*, *spike*, *right overlap*, *stressed right overlap*, *left and right overlap*, and *irregular overlap*. Spikelet is optimized to match a spike, called the *default spike*. The overlaps are classified based on their decomposing in two or more spikes.

The seven patterns illustrated in Fig. 5 cover almost all the spike and overlap shapes that were found in the corresponding signals. We sampled about 30 h of data, which generated spikes at a rate of about 40 Hz. We highlight that the spike shape means that only the H1 neuron action potential was captured by the electrode. This is an ideal situation, but it depends on perfect electrode positioning. So, when an overlap shape is captured, it means that another neuron (or neurons) is (are) reacting together (or not) with H1. The shape named *irregular overlap* is a deformed spike that conserves the ideal shape in its first part. At its end it appears irregular, indicating that a neighboring neuron is acting after H1, probably with a smaller amplitude. All the other overlap shapes indicate that the H1

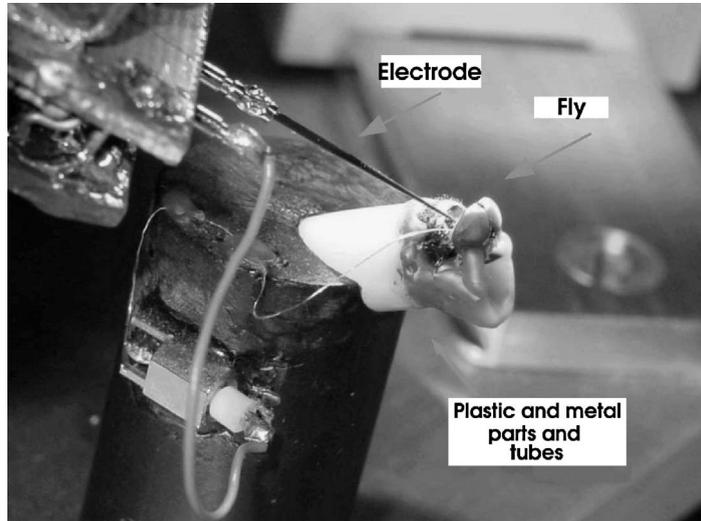


Fig. 1. The fly, the electrode, and the experiment.



Fig. 2. The Diptera lab.

reaction (associated with the largest amplitude in the shape) was preceded or followed by the reaction of another neighboring neuron(s).

As a final result, the instant that the spike (or overlap) appears correlates with the instant that some random bar appears. In this work, the instant the specified shape occurs is defined as the point in the shape with the greatest inclination, that is, the largest absolute value of its derivate in relation to time.

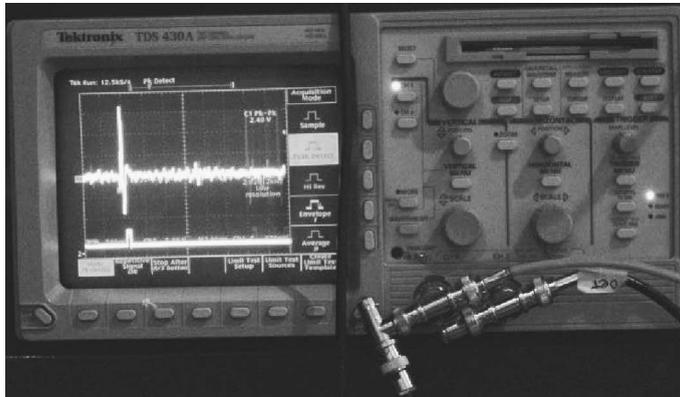


Fig. 3. A spike captured in real time.

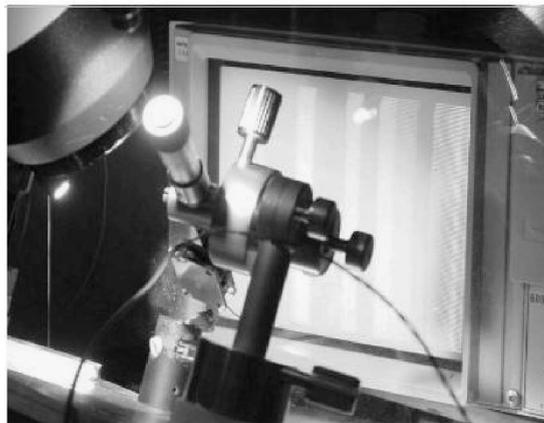


Fig. 4. The pattern of bars the fly sees during the experiment.



Fig. 5. 7 patterns of spikes and overlaps easily classified using Spikelet.

The behavior of the neurons, described by the instant the spikes and overlaps occur and the associated shapes, is very important in order to understand the behavior of the fly's visual system. The detection of neural spike and overlap activity is a technical challenge and is a prerequisite for studying many types of brain functions. The fly's implicit analog (stimulus velocity)-to-digital (spike and overlap trains) biological system can be used to study its neural code [10].

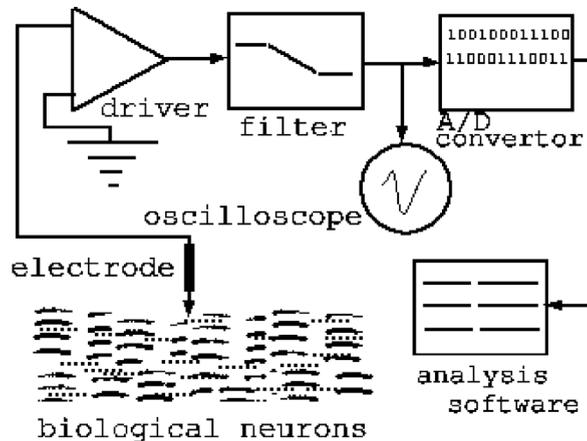


Fig. 6. Basic architecture for analyzing extracellular neural signals.

1.3. Some methods for spike and overlap sorting

The first link between neural communication and electrical signals was made by Luigi Galvani in 1791 when he showed that frog muscles could be stimulated by electricity [11]. Since the 1920s, nerve impulses have been measured directly by amplifying electrical signals recorded using electrodes, electronic amplifiers, and filters (Fig. 6).

There are some references in the literature that describe algorithms and methods for spike and overlap sorting [11–13]. The most significant are briefly related:

- **Threshold trigger algorithm.** This is the simplest method to detect a spike. A portion of a signal is classified as a spike whenever some of its samples cross a defined threshold. Although this method is the easiest to be implemented, it is very primitive because there is no shape classification, and overlaps cannot be found.
- **Threshold trigger algorithm with interspike interval histogram.** The threshold is used together with a histogram to measure the isolation among spikes that can generate overlaps. This method is also easy to implement, but the shape classification is not the best one. It checks the quality of interspike isolation over a long period of time. In cases where the spike waveform under analysis is a single unit, there should be no interspike interval less than the refractory period, which is of the order of 1 ms. This method requires a large number of spikes to be certain that it is an isolated unit.
- **Template matching algorithm.** In this case the algorithm tries to measure characteristics such as the height and width of the probable spike or overlap to get a better result on classification.
- **Principal component analysis algorithm.** The idea behind this method (Glaser and Marks, 1968; Glaser, 1971; and Gerstein et al., 1983) is to find an ordered set of orthogonal basis vectors that captures the directions of the largest variation in the data. According to the literature, it is one of the best methods for pattern classification.
- **Lewicki algorithm.** Lewicki (1994) introduced an overlap decomposition algorithm that uses a special data structure for pattern classification: time alignment and overlap

decomposition are done at the same time and with minimal computational expense. The algorithm uses k -dimensional search trees and returns a list of spike model sequences and their relative probabilities. The procedure runs fast enough to allow implementation in real time with only modest computing requirements.

- **Bayesian approach.** Rinberg et al. (2003) proposed a technique based on the Bayesian model of probability in the frequency-domain that efficiently detects spikes and overlaps and can be used in real time.
- **Neural networks approach.** A method based on learning which has a good performance in classification, although the results depend on the procedure used to train the classifiers. This approach customarily uses an algorithm of high-order complexity.
- **Filter-based methods.** This approach uses optimal filtering (Roberts and Hartline, 1975; Stein et al., 1979; Gozani and Miller, 1994) to discriminate a set of spikes from each other and from background noise. For a spike model, the filter is constructed to respond maximally to the spike shape of interest and minimally to the background noise by using convolution in the time domain. According to the literature mentioned above, the filter based methods to spike and overlap sorting are less effective than other methods such as principal component analysis, template matching, and some others in this classification. The use of Spikelet, that is, a wavelet transform, and thus, a filter, aims to change this idea significantly.

1.4. Time–frequency analysis review and discrete wavelet transform at work

The experiments with spike and overlap sorting algorithms have shown that it is possible to treat the problem in the frequency-domain. We present a brief review of the main techniques used to analyze a discrete-time signal both in time and frequency domains.

- **DFT/FFT.** The discrete Fourier transform (DFT) or, in its ‘fast’ version, fast Fourier transform (FFT), indicates the frequencies contained in a discrete-time signal, but not their occurrence in time. Thus, for a non-stationary signal, as the fly’s neural signal studied here, this technique does not work well. The DFT is

$$F[k] = \sum_{n=0}^{N-1} f[n]e^{-j2\pi nk/N}, \quad (1)$$

where k labels each sample of the DFT, $j = \sqrt{-1}$, and N is the number of samples of the original signal $f[n]$.

- **STFT.** The short-time Fourier transform (STFT), written in Eq. (2), which is a windowed version of DFT, makes a time–frequency analysis of a signal, but the window width may cause a problem of resolution: narrow windows give good time resolution but poor frequency resolution; wide windows give good frequency resolution but poor time resolution. In Eq. (2), $x(t)$ is the continuous original signal in the time domain, W is the window function, $j = \sqrt{-1}$, t is the time, and ω is the frequency:

$$\text{STFT}(t, x(t)) = \int_a^b (x(t)W(t-t'))e^{-j\omega t} dt. \quad (2)$$

- **DWT.** The discrete wavelet transform (DWT), whose main idea is the process of multiresolution analysis (MRA) proposed by Mallat [14], is one of the most appropriate techniques to make a joint time–frequency analysis of discrete-time signals. It allows one to find the time support of frequencies.

Considering \vec{f} as the discrete-time signal under analysis, it is decomposed in the sum of two other vectors \vec{A} and \vec{D} , called respectively *trend* and *fluctuation*,

$$\vec{f} = \vec{A} + \vec{D}, \quad (3)$$

where

$$\vec{A} = \sum_{k=0}^{n/2-1} \langle \vec{f}, \vec{v}_k \rangle \vec{v}_k, \quad \vec{D} = \sum_{k=0}^{n/2-1} \langle \vec{f}, \vec{w}_k \rangle \vec{w}_k.$$

Then, for a discrete-time signal \vec{f} containing n samples:

- \vec{A} is the projection of \vec{f} onto a subspace V , with a basis containing $n/2$ vectors;
- \vec{D} is the projection of \vec{f} onto a subspace W , with a basis containing $n/2$ vectors;
- $V \perp W \Leftrightarrow \vec{A} \perp \vec{D}$;
- $\vec{v}_i \perp \vec{w}_i \Leftrightarrow \langle \vec{v}_i, \vec{w}_i \rangle = 0$.

The process above is a one-level decomposition. When n is an integer power of 2, this process can be repeated $j = \log(n)/\log(2)$ times. To do that, the resulting signal \vec{A} is decomposed one or more times, creating a decomposition of j levels. This is the main idea behind the MRA analysis: decomposing a signal in several levels of resolution (Eq. (4)):

$$\vec{f} = \vec{A}_j + \sum_{i=1}^j \vec{D}_i. \quad (4)$$

Thus,

- \vec{A}_j is the projection of \vec{f} onto a subspace V_j , with a basis containing $n/2^j$ vectors;
- \vec{D}_i is the projection of \vec{f} onto a subspace W_j , with a basis containing $n/2^i$ vectors;
- $V_j \perp W_j \Leftrightarrow \vec{A}_j \perp \vec{D}_j$;
- $\vec{v}_{i,j} \perp \vec{w}_{i,j} \Leftrightarrow \langle \vec{v}_{i,j}, \vec{w}_{i,j} \rangle = 0$.

In other words,

$$f[n] = \sum_{k=0}^{n/2^j-1} H_{j,k}[n] \phi_{j,k}[n] + \sum_{t=1}^j \sum_{k=0}^{n/2^t-1} G_{t,k}[n] \psi_{t,k}[n], \quad (5)$$

where

- $\phi[n]$ and $\psi[n]$ forms a Riezs basis [15] to write signal f ;
- $\phi[n] = \sum_k h_n \phi[2n - k]$, defined recursively by dilations and translations of itself, is called *scaling function* [15];

- $\psi[n] = \sum_k g_n \phi[2n - k]$, defined recursively, is called *wavelet function* and is orthogonal to the *scaling function*;
- $H_{j,k}[n] = \langle f, \phi_{j,k}[n] \rangle$;
- $G_{i,k}[n] = \langle f, \psi_{i,k}[n] \rangle$;
- $\{0\} \leftarrow \dots \subset V_{-1} \subset V_0 \subset V_1 \subset \dots \rightarrow L^2$;
- if $f[n] \in V_j \rightarrow f[2n] \in V_{j+1}$;
- $V_{j+1} = V_j \oplus W_j$;
- the h_k coefficients form a lowpass filter;
- the g_k coefficients form a highpass filter;
- the h_k and g_k coefficients form the *analysis filter bank*;
- a filter with k coefficients is called a filter of support k .

In practice, to compute the DWT, the coefficients of the signal under analysis (f) are convolved [19] both with the lowpass filter (h) and the highpass (g) filter as follows:

$$y_{\text{lowpass}}[n] = f[\] * h[\] = \sum_{k=0}^{M-1} h[k] f[2n - k], \quad 0 \leq n \leq N/2, \quad (6)$$

$$y_{\text{highpass}}[n] = f[\] * g[\] = \sum_{k=0}^{M-1} g[k] f[2n - k], \quad 0 \leq n \leq N/2, \quad (7)$$

where y_{lowpass} and y_{highpass} are the outputs, M is the length of signals h and g , N is the length of signal f , and $*$ denotes the discrete-time convolution.

In each level of decomposition, there is a downsampling by 2 of the signals being analyzed because the convolutions above are, in fact, filters that allow a half-band of the original signal to pass, according to *Nyquist rate* [17]. Figure 7 illustrates this process.

A much more detailed description of the DWT working, including the inverse DWT, can be found in the literature, especially [1,6,15]. It is not the goal of this work to detail all of the properties, but there are three special properties of a wavelet transform that are used and analyzed in Spikelet. They are:

- **Number of vanishing moments.** To a wavelet with m vanishing moments and a signal that can be described approximately by a polynomial of degree less than m , the fluctuation (detail) coefficients are approximately zero. Each moment can be calculated with Eq. (8), where P is the number of integer and half points of the wavelet function that are different from zero, $m = 0, 1, 2, \dots$, is the desired moment, $t_0 = 0$, $t_1 = 0.5$, $t_2 = 1$, $t_3 = 1.5$, and so on. The number of vanishing moments is thus defined as the non-negative integer $m - 1$ for which the (right-hand side) (rhs) of Eq. (8) differs from zero:

$$\text{moment}_m = \sum_{k=0}^{P-1} t_k^m \psi[t_k]. \quad (8)$$

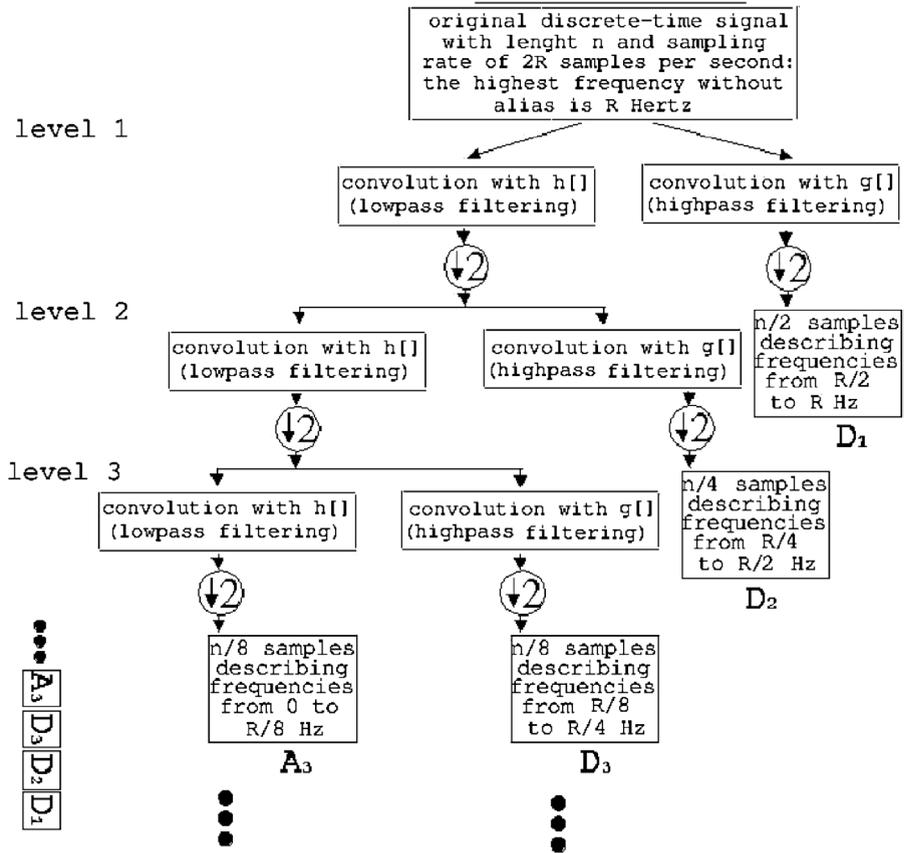


Fig. 7. Example of a MRA analysis using DWT in a 3-level decomposition. The DWT of the original signal with n samples also has n samples. In the higher frequencies, there is better time resolution and worse frequency resolution; in the lower frequencies, the opposite occurs.

- **Signal energy**, defined to be the scalar value E in Eq. (9),

$$E = \sum_{n=0}^{N-1} f[n]^2, \quad (9)$$

where N is the length of signal f .

- **QMF filters** (quadrature mirror filters), corresponding to the pair of digital filters defined by h_0, h_1, \dots, h_k and g_1, g_2, \dots, g_k coefficients, where $\{h_k\}$ form a lowpass filter and $\{g_k\}$ form the mirror highpass filter [16], in terms of its frequency response, according to Eq. (10),

$$g_k = (-1)^k h_{M-k-1}, \quad (10)$$

where M is the support length, or the number of coefficients.

1.5. Methods to create wavelets matching a signal

In general, wavelet transforms such as Daubechies, Coiflets, Symmlets [1], and others are not optimized to match a specified signal. Among the methods, two of the most interesting and recent ones that consider the signal being analyzed are briefly described below:

- **Closed-form solutions for finding scaling function spectrum from the wavelet spectrum.** Proposed by Chapa and Rao [5], the wavelets are designed directly using the signal to match instead of building a composite wavelet from a library of previously designed wavelets. The basis is then modified. To do that, two sets of equations are created. This imposes band limitations and results in closed-form solutions. The algorithm generates the associated scaling function which, in turn, satisfies the orthonormal multi-resolution analysis criteria.
- **Maximal projection of given signal on to the scaling subspace.** In this method, proposed by Gupta et al. [7], the signal to be matched is convolved with a generic lowpass filter with support 4 or 6. The resulting signal energy on the scaling subspace is then maximized by calculating its first derivative in relation to all but one filter weight. This yields a linear homogeneous system. The j th filter weight is then kept constant to 1, and the resulting system is solved to obtain the other filter coefficients. The problem with this method is that the result depends on the discarded filter weight. This idea corresponds to the energy minimization of a signal that moves to consecutive wavelet subspaces. Thus, the projection on the scaling subspace is maximized. The technique proposed here to construct Spikelet makes use of this idea, but it does not discard any filter weight.

2. The default spike as the signal to match

The default spike, the main signal to be matched in this instance, is defined using an experimental process. It is the middle spike among the 80 spikes that were obtained from the visual inspection of a large signal containing many spikes and overlaps. Figure 8 shows the default spike together with the numerical values ($x = \text{sample}$, $y = \text{amplitude}$) for each one of its 32 points. The length of the signal to match is arbitrary: 32 is merely the example used in this case.

To better illustrate the patterns to match, Fig. 9 shows one clip of a recorded signal of interest, both in time and frequency domain. Some isolated spikes and overlaps that are decomposed in several frequency sub-bands are shown in Figs. 10–13 to give an idea about the shapes and the frequencies that form these patterns.

3. Construction of the Spikelet wavelet transform

The process to create the new wavelet, Spikelet, consists of obtaining its filter coefficients using the least-squares method to solve a system of $3n/2 + 1$ linear equations in n

(0, 4533)	(1, 5562)
(2, 6517)	(3, 7358)
(4, 7858)	(5, 7278)
(6, 4903)	(7, 187)
(8, -6648)	(9, -14330)
(10, -21331)	(11, -26532)
(12, -29400)	(13, -29865)
(14, -28071)	(15, -24728)
(16, -20572)	(17, -15930)
(18, -11207)	(19, -6682)
(20, -2644)	(21, 807)
(22, 3627)	(23, 5771)
(24, 7318)	(25, 8302)
(26, 8771)	(27, 8772)
(28, 8421)	(29, 7822)
(30, 7044)	(31, 6222)

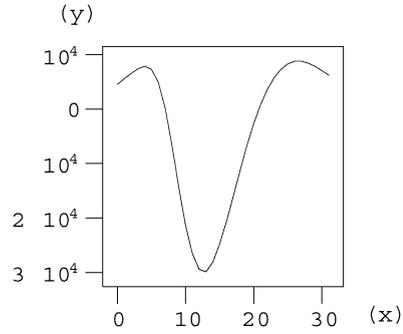


Fig. 8. Default spike.

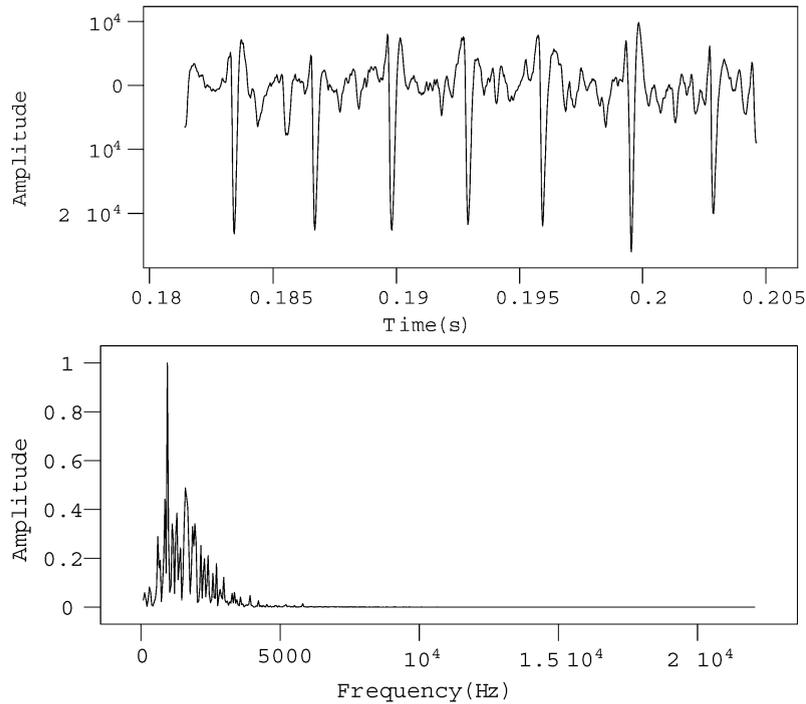


Fig. 9. (Top) Some spikes and overlaps in time domain captured during the experiment. (Bottom) The corresponding frequency domain representation.

unknowns, n being the filter support. These equations come from two existing approaches to construct a wavelet, both suitably modified:

- ordinary Daubechies wavelet construction [6], excluding the orthogonality criteria. This technique does not consider the signal being analyzed;

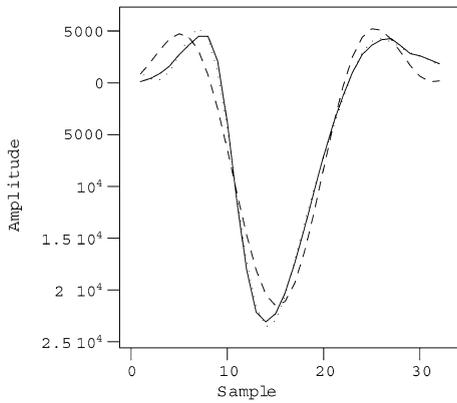


Fig. 10. Spike and lowpass filters: solid, original; dashes, 5512.5 Hz; dots, 2756.25 Hz.

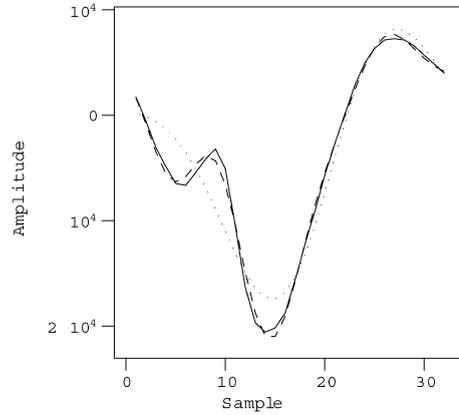


Fig. 11. Overlap and lowpass filters: solid, original; dashes, 5512.5 Hz; dots, 2756.25 Hz.

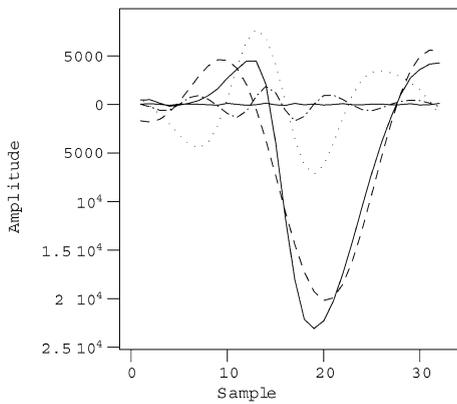


Fig. 12. Spike and bandpass filters: solid, original; dashes, 0–2756.25 Hz; dots, 2756.25–5512.50 Hz; dashes and dots, 5512.50–11025 Hz; dash with three dots, 11025–22050 Hz.

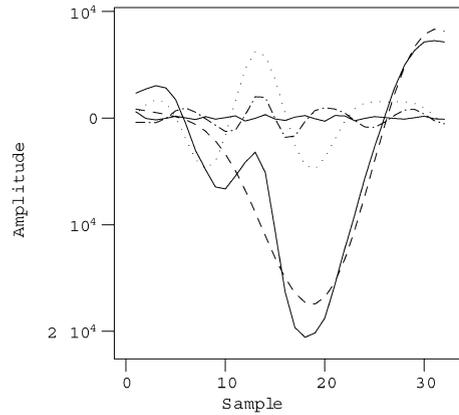
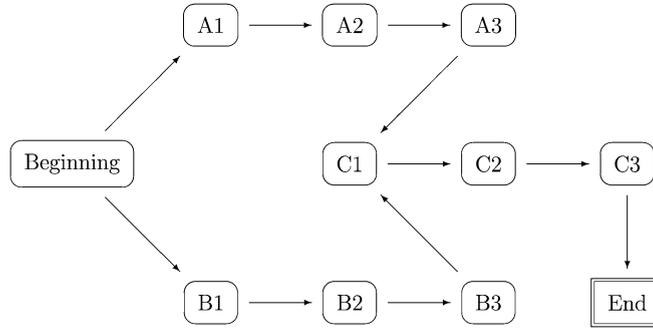


Fig. 13. Overlap and bandpass filters: solid, original; dashes, 0–2756.25 Hz; dots, 2756.25–5512.50 Hz; dashes and dots, 5512.50–11025 Hz; dash with three dots, 11025–22050 Hz.

- maximization of the signal to match on a scaling subspace when its DWT is performed [7], excluding no filter weight.

The full description follows. Once the signal to match, default spike, has been defined, three sequences of steps have to be followed: sequence A (A1, A2, and A3), sequence B (B1, B2, and B3), which may happen in parallel with sequence A, and finally, sequence C (C1, C2, and C3), which occurs after sequences A and B have been completed. This process is illustrated in the following diagram, followed by a detailed description of A, B, and C:



- **Step A1.** Write down the equations for the Daubechies transform filter coefficients h_i :

$$\sum_{k=0}^{n-1} (-1)^k h_k k^b = 0, \quad \sum_{k=0}^{n-1} h_k = 2, \quad \sum_{k=0}^{n-1} h_k h_{k+2l} = 2\delta_{0,l},$$

where $b = 0, 1, \dots, n/2 - 1$ and $l \in \mathbb{Z}$. The first equation is responsible for producing vanishing moments. The second one comes from the fact that the area under the scaling function is unitary: $\int \phi(x) dx = 1$. Finally, the third one imposes orthogonality among the scaling and wavelet vectors. This is simply the original Daubechies transform.

- **Step A2.** The condition of orthogonality is excluded. The Spikelet transform will not be orthogonal:

$$\sum_{k=0}^{n-1} (-1)^k h_k k^b = 0, \quad \sum_{k=0}^{n-1} h_k = 2.$$

This does not represent a problem because no inverse transform is needed in the algorithm proposed to classify the patterns. Although the spike and overlap signals Spikelet proposes to match are not smooth, the first group of equations in step A2, $\sum_{k=0}^{n-1} (-1)^k h_k k^b = 0$, responsible for producing the vanishing moments, are kept in order to ensure the right frequency response of the proposed filters.

- **Step A3.** Support $n = 4$ is imposed on the equations from step 2. It produces a system of $n/2 + 1 = 3$ linear equations in $n = 4$ unknowns:

$$\begin{cases} -1h_3 + 1h_2 - 1h_1 + 1h_0 = 0, \\ -3h_3 + 2h_2 - 1h_1 + 0h_0 = 0, \\ 1h_3 + 1h_2 + 1h_1 + 1h_0 = 2. \end{cases}$$

Constructing the transform with larger supports worsens the result in terms of signal recognition, as the experiments have shown.

- **Step B1.** The default spike signal is convolved with a generic signal of support $n = 4$ in the unknowns h_0, \dots, h_3 , followed by a downsampling by 2. The resulting signal is the projection of default spike, called vector \vec{u} , on a subspace V .
- **Step B2.** The resulting signal energy of step B1 is maximized:

$$\frac{\partial E(\overline{P_V \vec{u}})}{\partial h_0} = 0, \quad \frac{\partial E(\overline{P_V \vec{u}})}{\partial h_1} = 0, \quad \frac{\partial E(\overline{P_V \vec{u}})}{\partial h_2} = 0, \quad \frac{\partial E(\overline{P_V \vec{u}})}{\partial h_3} = 0.$$

- **Step B3.** Step B2 leads to a system of linear homogeneous equations in $n = 4$ unknowns:

$$\begin{cases} -3.4984h_3 + 3.3876h_2 - 3.1079h_1 + 2.6457h_0 = 0, \\ -3.3876h_3 + 3.4876h_2 - 3.3967h_1 + 3.0924h_0 = 0, \\ -3.1079h_3 + 3.3967h_2 - 3.5100h_1 + 3.4000h_0 = 0, \\ -2.6457h_3 + 3.0924h_2 - 3.4000h_1 + 3.5006h_0 = 0. \end{cases}$$

- **Step C1.** The equations of steps A3 and B3 are grouped together, resulting in a system of $(n/2 + 1) + n = 3n/2 + 1 = 7$ linear equations in $n = 4$ unknowns:

$$\begin{pmatrix} -3.4984 & 3.3876 & -3.1079 & 2.6457 \\ -3.3876 & 3.4876 & -3.3967 & 3.0924 \\ -3.1079 & 3.3967 & -3.5100 & 3.4000 \\ -2.6457 & 3.0924 & -3.4000 & 3.5006 \\ -1 & 1 & -1 & 1 \\ -3 & 2 & -1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}.$$

- **Step C2.** The system is solved using the least-squares method:

$$\begin{pmatrix} -3.4984 & -3.3876 & -3.1079 & -2.6457 & -1 & -3 & 1 \\ 3.3876 & 3.4876 & 3.3967 & 3.0924 & 1 & 2 & 1 \\ -3.1079 & -3.3967 & -3.5100 & -3.4000 & -1 & -1 & 1 \\ 2.6457 & 3.0924 & 3.4000 & 3.5006 & 1 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} -3.4984 & 3.3876 & -3.1079 & 2.6457 \\ -3.3876 & 3.4876 & -3.3967 & 3.0924 \\ -3.1079 & 3.3967 & -3.5100 & 3.4000 \\ -2.6457 & 3.0924 & -3.4000 & 3.5006 \\ -1 & 1 & -1 & 1 \\ -3 & 2 & -1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}.$$

- **Step C3.** The solution h_0, \dots, h_3 forms a lowpass filter. Defined as a QMF system, the corresponding highpass filter is formed by the g_0, \dots, g_3 coefficients according to Eq. (10):

$$\begin{aligned} h_0 &= 0.26964482896235847376, & g_0 &= 0.23524044702452745481, \\ h_1 &= 0.76237548312490721614, & g_1 &= -0.73270322306815560687, \\ h_2 &= 0.73270322306815560687, & g_2 &= 0.76237548312490721614, \\ h_3 &= 0.23524044702452745481, & g_3 &= -0.26964482896235847376. \end{aligned}$$

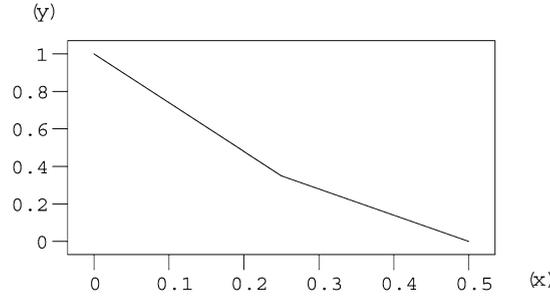


Fig. 14. Frequency response of the Spikelet lowpass filter, where x represents the angular frequency ($\pi = 0.5$) and y represents the amplitude.

Figure 14 shows the frequency response of the Spikelet lowpass filter. Although the frequency response of the Daubechies lowpass filter with support 4 is closer to an ideal response, the Spikelet transform tries to match a signal, not just filter it. On the other hand, the frequency response cannot be arbitrary because in this case the wavelet transform of the signal under analysis would not relate to the expected frequencies in each level of decomposition. Thus, the Spikelet filters show a satisfactory frequency response.

The next two sections show how to estimate the corresponding scaling and wavelet functions of Spikelet. They are not necessary in order to use the proposed algorithm because only an analysis (decomposition) of the input signal is made, and no inverse transform (synthesis) is needed.

3.1. Scaling function

The scaling function, $\phi(x)$, defined recursively using the dilation equation $\phi(n) = \sum_k h_k \phi(2n - k)$ for a system of support $n = 4$ coefficients, does not exist to $x < 0$ and to $x > 3$. We therefore get

$$\begin{cases} \phi(0) = h_0 \phi(0), \\ \phi(1) = h_0 \phi(2) + h_1 \phi(1) + h_2 \phi(0), \\ \phi(2) = h_1 \phi(3) + h_2 \phi(2) + h_3 \phi(1), \\ \phi(3) = h_3 \phi(3), \end{cases}$$

or $MT = T$, where

$$M = \begin{pmatrix} h_0 & 0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 \\ 0 & h_3 & h_2 & h_1 \\ 0 & 0 & 0 & h_3 \end{pmatrix} \quad \text{and} \quad T = \begin{pmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \end{pmatrix}.$$

So, matrix T with scaling function values is the eigenvector of M corresponding to eigenvalue 1. Using the normalizing condition $\sum_k \phi(k) = 1$, we get

$$\begin{cases} (h_0 - 1)\phi(0) = 0, \\ h_2 \phi(0) + (h_1 - 1)\phi(1) + h_0 \phi(2) = 0, \\ h_3 \phi(1) + (h_2 - 1)\phi(2) + h_1 \phi(3) = 0, \\ (h_3 - 1)\phi(3) = 0, \\ \phi(0) + \phi(1) + \phi(2) + \phi(3) = 1. \end{cases}$$

As $h_0, \dots, h_3 \neq 0$,

$$\begin{cases} \phi(0) = \phi(3) = 0, \\ (h_1 - 1)\phi(1) + h_0\phi(2) = 0, \\ h_3\phi(1) + (h_2 - 1)\phi(2) = 0, \\ \phi(0) + \phi(1) + \phi(2) + \phi(3) = 1, \end{cases}$$

and so

$$\begin{cases} -0.23762451687509278386\phi(1) + 0.26964482896235847376\phi(2) = 0, \\ 0.23524044702452745481\phi(1) - 0.26729677693184439313\phi(2) = 0, \\ \phi(1) + \phi(2) = 1. \end{cases}$$

Using the least-squares method again,

$$\phi(1) = 0.53172639832546342298,$$

$$\phi(2) = 0.46827358753833858707,$$

whereas the intermediate points satisfy $\phi(x/2) = \sum_k h_k \phi(x - k)$. Thus,

$$\phi\left(\frac{1}{2}\right) = h_0\phi(1) = 0.14337727373124048436,$$

$$\phi\left(\frac{3}{2}\right) = h_1\phi(2) + h_2\phi(1) = 0.74659794837766346731,$$

$$\phi\left(\frac{5}{2}\right) = h_3\phi(2) = 0.11015688806229795293.$$

3.2. Wavelet function

Using the equation $\psi(n) = \sum_k g_k \phi(2n - k)$, we get

$$\begin{cases} \psi(0) = g_0\phi(0), \\ \psi(1) = g_0\phi(2) + g_1\phi(1) + g_2\phi(0), \\ \psi(2) = g_1\phi(3) + g_2\phi(2) + g_3\phi(1), \\ \psi(3) = g_3\phi(3), \end{cases}$$

yielding

$$\psi(0) = \psi(3) = 0,$$

$$\psi(1) = g_0\phi(2) + g_1\phi(1) = -0.27944075778119104037,$$

$$\psi(2) = g_2\phi(2) + g_3\phi(1) = 0.21362302880293393414,$$

and for the intermediary points, $\psi(x/2) = \sum_k g_k \phi(x - k)$. Thus,

$$\psi\left(\frac{1}{2}\right) = g_0\phi(1) = 0.12508355563682396761,$$

$$\psi\left(\frac{3}{2}\right) = g_1\phi(2) + g_2\phi(1) = 0.06226960294661321171,$$

$$\psi\left(\frac{5}{2}\right) = g_3\phi(2) = -0.12626755141936529814.$$

Figure 15 shows both the scaling and wavelet function graphs.

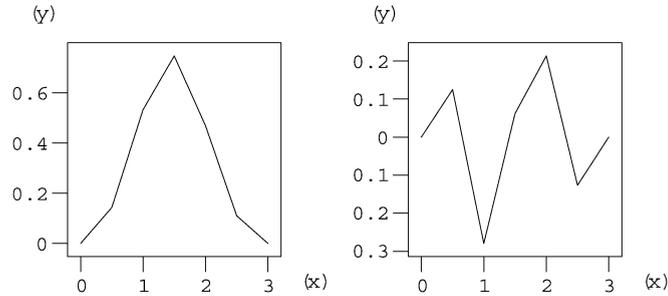


Fig. 15. Scaling (left) and wavelet (right) functions of the Spikelet transform.

3.3. Vanishing moments

Although the system in step C1 preserves both vanishing moment conditions for Spikelet, the least-squares method used in step C2 disturbs that somewhat. These moment conditions are, according to Eq. (8) for $t = 0, 0.5, 1, \dots, 2.5, 3$:

- 0-order moment = -0.0047321 ,
- 1-order moment = -0.0119173 .

Thus, in theory, Spikelet does not respect any moment condition. However, since it is very close to having two moment conditions satisfied, and the moment conditions are kept in step A2, we still classify it as a wavelet transform. Anyway, vanishing moments are not used by the algorithm for spike sorting proposed in Section 4.

3.4. Causality and stability

The pole-zero plot [18] of Spikelet lowpass filter shown in Fig. 16 indicates that it is a causal and stable system according to its transfer function:

$$H(z) = \frac{h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3}}{z^{-n}}$$

All the poles are located inside the unit circle, and although there are two zeros outside the circle, only the inverse system, not used or calculated here, may not be causal or unstable.

3.5. Considerations

For all tests, a double-numerical precision has been used to compute the filter coefficients. The Spikelet filter has the following characteristics:

- asymmetrical wavelet;
- FIR (finite impulse response) filter [19];
- compact support (4);
- stable and causal;

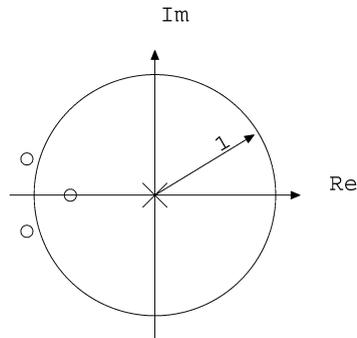


Fig. 16. Pole-zero plot.

- non-linear phase response;
- each vector X_j is only orthogonal to the corresponding vector B_j , $j = 0, 1, \dots$, where $\{X\}$ are the vectors of support 4 with coefficients h_0, \dots, h_3 that come from the scaling function, and $\{B\}$ are the ones that come from the wavelet function. The system is, by construction, not orthogonal;
- a 3-level decomposition of a spike with Spikelet shows that sample 2 presents the largest value of the transform among the samples 0 to 3. Sample 2 of the transform relates to the rising edge of the original spike. In the next section we see why this is so important [1, p. 232].

4. Using Spikelet: the proposed algorithm

The algorithm that uses Spikelet has a linear order of complexity and is based on an elementary analysis of signal variation between neighboring transform samples of the signal under analysis. The steps are:

- **Beginning**
- **Step 1.** The FFT (fast Fourier transform) of the signal under analysis is used to find the frequencies that compose it. The signals that were sampled at 44,100 kHz with 16 bits of quantization contained a higher frequency of 22,050 kHz without alias. From Nyquist's sampling theorem, the highest frequency is about 5512.50 kHz when the signal contains overlaps, and about 2756.25 kHz when only the spikes are present. Thus, a 3-level Spikelet decomposition was chosen.
- **Step 2.** To reduce computation time, a threshold trigger is used to detect spikes (or overlaps) that cross a certain pre-defined threshold. For each sample of the original signal that passes this test, a rectangular window of 32 samples is formed around this possible spike/overlap. This small part of the signal is analyzed using Spikelet (step 3). In case the test fails, the algorithm advances some samples in the original signal and returns to step 2 until the signal ends.
- **Step 3.** The 3-level Spikelet decomposition of the small windowed signal containing 32 samples is formed and analyzed as follows:

- inversion between samples 1 and 2, 4 and 5 → stressed left overlap;
 - inversion between samples 1 and 2, 4 and 5, 5 and 6 → left overlap;
 - inversion between samples 1 and 2, 4 and 5, 5 and 6, 6 and 7 → spike;
 - inversion between samples 1 and 2, 5 and 6, 6 and 7 → right overlap;
 - inversion between samples 1 and 2, 6 and 7 → stressed right overlap;
 - inversion between samples 1 and 2, 4 and 5, 6 and 7 → left and right overlap;
 - any other combination of signal variations → irregular overlap.
- **Step 4.** Some points are advanced. The signal returns to step 2 if it has not ended.
 - **End**

Note that samples 0–3 of the transform contain frequencies from 0 to 2756.25 kHz and, thus, describe only the spikes. Samples 4–7 contain the frequencies from 2756.25 to 5512.50 kHz and, thus, describe the overlaps.

Each inversion between samples i and $i + 1$, $0 \leq i \leq 6$, in step 3 above relates to a rising edge in the pattern being analyzed. According to Figs. 10–13, it is possible to see when each one of the 7 patterns decomposes in the corresponding Spikelet sub-bands.

Once a spike or an overlap is found, the associated “instant of time” that marks its occurrence is defined as the point in which the derivative is maximized.

5. Results and conclusion

Spikelet easily detects and classifies the rising edges of spikes and the overlaps in the corresponding frequency sub-bands. In particular, no other wavelet transform, probabilistic method, or deterministic method (as this algorithm is) is able to detect and recognize these patterns. Table 1 summarizes the results, comparing Spikelet with the proposed algorithm used together with other wavelet transforms and also with other algorithms and approaches that have been discussed earlier.

Filter-based methods are the ones that present the worst results to spike sorting [11]. Our technique changes and innovates this.

The Spikelet transform and the algorithm for pattern recognition can be adapted to different kinds of signals, including electroencephalogram (EEG) signals, speech signals, and others. The algorithm using Spikelet was implemented using C++ programming language under a LINUX operating system in a personal computer and in a digital signal processor, ADSL21065L, SHARC, Analog Devices Inc., for use in real time.

An interesting open discussion: As the filter nears the ideal, the pattern classification worsens. This is visible when testing the wavelets above using the same algorithm. A more selective filter, one with a frequency response closer to the ideal, is not the best for selecting patterns.

At the moment, the same idea used to create Spikelet is being tested to create a matched wavelet used to sort and classify real time human voice disorders [20].

Please send questions and comments directly to the corresponding author.

Table 1
Comparison: Spikelet and other spike sorting methods

Method ^a	Algorithm complexity	Deterministic (D)/ probabilistic (P)	Number of patterns discovered
Threshold trigger	Linear	D	1
Threshold + histogram	Linear	P	1–2
Matching template	Linear	D	2–4
Principal component	Linear	D	2–4
Bayesian	Linear	P	2–4
Lewicki	Not linear	D	2–4
Neural networks	Variable	P	Variable ^b
Generic filter based	Variable	D/P	2–4
Daubechies sp 20*	Linear	D	2
Symmlet sp 8*	Linear	D	3
Daubechies sp 4*	Linear	D	4
Haar (sp 2)*	Linear	D	4
Beylkin sp 18*	Linear	D	4
Vaidyanathan sp 24*	Linear	D	4
Coiflet sp 6*	Linear	D	5
Battle–Lemariè sp 12*	Linear	D	5
Spikelet (sp 4)*	Linear	D	7

^a sp = support size; * = DWT used with proposed algorithm.

^b 6 or 7 patterns can be recognized, though, only by using a complexity network of high order.

Acknowledgments

We thank the researchers from FFI-IFSC-USP, Instituto de Física de São Carlos da Universidade de São Paulo, Brazil (<http://www.if.sc.usp.br>) and from DCCE-IBILCE-UNESP, Departamento de Ciência da Computação e Estatística, Universidade Estadual Paulista, Brazil (<http://www.dcce.ibilce.unesp.br>), especially Professor Dr. José Márcio Machado and Professor Dr. Eliana Xavier Linhares de Andrade. We wish to thank Analog Devices Inc. for their generous donation of the DSP ADSL21065L-SHARC used in the experiment. We are very grateful to Patricia Marie Toth-Interlando for all her help with the English language.

References

- [1] P.S. Addison, *The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance*, Institute of Physics Publishing, Edinburgh, 2002.
- [2] R.A. Gopinath, J.E. Odegard, C.S. Burrus, Optimal wavelet representation of signals and wavelet sampling theorem, *IEEE Trans. Circuit Syst. II* 41 (1994) 262–277.
- [3] M. Akay, *Time Frequency and Wavelets in Biomedical Signal Processing*, IEEE Press in Biomedical Engineering, New York, 1998.
- [4] C. Fung, P. Shi, Design of compactly supported wavelet to match singularities in medical images, *Appl. Digital Image Process.* XXV (2002) 358–369.
- [5] J.O. Chapa, R.M. Rao, Algorithms for designing wavelets to match a specified signal, *IEEE Trans. Signal Process.* 4 (12) (2000) 3395–3406.

- [6] J.R. Williams, K. Amaratunga, Introduction to wavelets in engineering, *Int. J. Numer. Methods Eng.* 37 (1994) 2365–2388.
- [7] A. Gupta, S.D. Joshi, S. Prasad, On a new approach for estimating wavelet matched to signal, in: *Proceedings 8th National Conference on Communications*, IIT, Bombay, 2002, pp. 180–184.
- [8] S.S. Haykin, B. Widrow, *Least-Mean-Square Adaptive Filters*, Wiley, Indianapolis, 2003.
- [9] R.R.R.V. Steveninck, G.D. Lewen, S.P. Strong, R. Koberle, W. Bialek, Reproducibility and variability in neural spike trains, *Science* 275 (1997) 1805–1808.
- [10] F. Rieke, W. Bialek, D. Warland, R.R. Steveninck, *Spikes: Exploring the Neural Code*, first ed., MIT Press, 1999.
- [11] M.S. Lewicki, A review of methods for spike sorting: the detection and classification of neural action potentials, *Network Comput. Neural Syst.* 9 (1998) R53–R78.
- [12] K.M.L. Menne, A.M.A.T. Folkers, R. Maex, U.G. Hofmann, Test of spike sorting algorithms on the basis of simulated network data, *Neurocomputing* 44–46 (2002) 1119–1126.
- [13] D. Rinberg, W. Bialek, H. Davidowitz, N. Tishby, Spike sorting in the frequency-domain with overlap detection, Preprint, NEC Research Institute, 2003, pp. 1–40; available at <http://arxiv.org/abs/physics/0306056>.
- [14] S.G. Mallat, *Multiresolution approximation and wavelets*, Preprint, GRASP Laboratory, Department of Computer and Information Science, University of Pennsylvania, 1986.
- [15] G. Strang, T. Nguyen, *Wavelets and Filter Banks*, Wellesley–Cambridge Press, Wellesley, 1997.
- [16] R.W. Hamming, *Digital Filters*, third ed., Dover, New York, 1989.
- [17] J. Lane, J. Datta, B. Karley, J. Norwood, *DSP Filters*, Prompt Publications, Indianapolis, 2001.
- [18] S. Haykin, B.V. Veen, *Signals and Systems*, second ed., Wiley, Indianapolis, 2002.
- [19] A.V. Oppenheim, R.W. Schaffer, *Discrete-Time Signal Processing*, second ed., Prentice Hall, New York, 1999.
- [20] R.C. Guido, J.C. Pereira, E. Fonseca, L.S. Vieira, F.L. Sanchez, Trying different wavelets on the search for voice disorders sorting, in: *Proceedings of the 37th IEEE SSST—International Southeastern Symposium on System Theory*, Tuskegee, AL, vol. 1, IEEE Press, 2005, pp. 459–499.