ELSEVIER

Journal of Neuroscience Methods 101 (2000) 93–106

JOURNAL OF NEUROSCIENCE METHODS

www.elsevier.com/locate/jneumeth

# Spike sorting based on discrete wavelet transform coefficients

Juan Carlos Letelier *, Pamela P. Weber

*Departamento de Biología, Facultad de Ciencias, Universidad de Chile, Casilla 653, Santiago, Chile*

Received 16 February 2000; received in revised form 8 May 2000; accepted 22 May 2000

**Abstract**

Using the novel mathematical technique known as wavelet analysis, a new method (WSC) is presented to sort spikes according to a decomposition of neural signals in the time–frequency space. The WSC method is implemented by a pyramidal algorithm that acts upon neural signals as a bank of quadrature mirror filters. This algorithm is clearly explained and an overview of the mathematical background of wavelet analysis is given. An artificial spike train, especially designed to test the specificity and sensibility of sorting procedures, was used to assess the performance of the WSC method as well as of methods based on principal component analysis (PCA) and reduced feature set (RFS). The WSC method outperformed the other two methods. Its superior performance was largely due to the fact that spike profiles that could not be separated by previous methods (because of the similarity of their temporal profile and the masking action of noise) were separable by the WSC method. The WSC method is particularly noise resistant, as it implicitly eliminates the irrelevant information contained in the noise frequency range. But the main advantage of the WSC method is its use of parameters that describe the joint time–frequency localization of spike features to build a fast and unspecialized pattern recognition procedure. © 2000 Elsevier Science B.V. All rights reserved.

*Keywords:* Spike-sorting; Spike-classification; Wavelet; Filter bank algorithm; DWT; WSC

## 1. Introduction

The separation of action potentials (spikes), when more than one neuron is recorded with a single microelectrode, presents a common problem in neurophysiology. As the microelectrode tip is surrounded by many neurons, it detects the occurrence of the electrical events generated by all the neurons in the area. Usually the experimenter tries to optimize the recording situation, so as to enhance the response of only one neuron. Unfortunately, experimental techniques are limited in achieving clear isolated recordings. Therefore, to increase the data yield of each experiment, spike sorting is a crucial step to extract coherent signals of a single target neuron from this mixture of responses.

The spike-sorting problem has received intense attention, and many partial or ad-hoc solutions have been advanced. These solutions range from purely electronic methods, like window discriminators (Schmidt, 1984a) or the use of tetrodes (Gray et al., 1995), to various numerical procedures applied off-line to the data stream. Numerical techniques span a continuum of approaches (Schmidt, 1984b), from the very simple discrimination by amplitude, the use of principal component analysis (PCA) (Eggermont et al., 1983), reduced feature set (RFS) (Salganicoff et al., 1988; Sarna et al., 1988), to neural networks (Oghalai et al., 1994) and even specialized commercial hardware implementing on-line pattern matching (http://www.sigprosys.com.au). All sorting or classification algorithms extract from the set of digital samples, which represent the profile of each spike, a small number ($m$) of parameters and use them to represent each spike in an $m$-dimensional space ($S_m$). Numerical separation is achieved as some parameters produce clusters of points in $S_m$. Ideally, each cluster contains the spikes produced by one and only one neuron. The performance of all sorting algorithms is reduced, in different degrees, by the unavoidable presence of noise in extracellular recordings.

Here we present another method, called the Wavelet-based Spike Classifier (WSC) method, based on the quantification of energy found in specific frequency

* Corresponding author. Tel.: +56-2-6787365; fax: +56-2-2712983.

*E-mail address:* letelier@uchile.cl (J.C. Letelier).

0165-0270/00/$ - see front matter © 2000 Elsevier Science B.V. All rights reserved.
PII: S0165-0270(00)00250-8

bands at specific time locations during each spike profile. The core idea of the WSC method rests upon the observation that differences between profiles primarily amount to transient differences in high frequency features (like sharp edges and steep leading or trailing slopes) and/or in low frequency features (like the duration of the repolarization phase). To obtain this localized time–frequency information we perform a *discrete wavelet analysis* upon the temporal profile of each detected spike. Wavelet analysis is a novel mathematical tool (Meyer, 1993; Burrus et al., 1997; Chui, 1997) that, like Fourier analysis, represents signals with functions that are bounded in frequency, but contrary to Fourier analysis these functions are also bounded in time (Fig. 1). This latter characteristic makes wavelet analysis increasingly useful in a variety of fields (Burrus et al., 1997; Chui, 1997), including the neurosciences (for a review see Samar et al., 1999). We recommend the following references as introductory work on wavelet analysis and its applications (Meyer, 1993; Strang, 1993; Aldroubi and Unser, 1996; Burrus et al., 1997; Samar et al., 1999).

Because the wavelet decomposition of signals is a rather recent technique, this paper is arranged as follows: a short general introduction to wavelet analysis, an explanation of the mathematical steps involved in calculating the discrete wavelet transform of digitized data, presentation of the WSC method, application of the WSC method to an artificial spike train, comparison with PCA and RFS sorting procedures, and finally discussion of the advantages and disadvantages of using wavelet analysis in recognition of signal profiles. All numerical procedures were coded using IGOR 3.12 (www.wavemetrics.com) in a Win NT 4.1 environment.

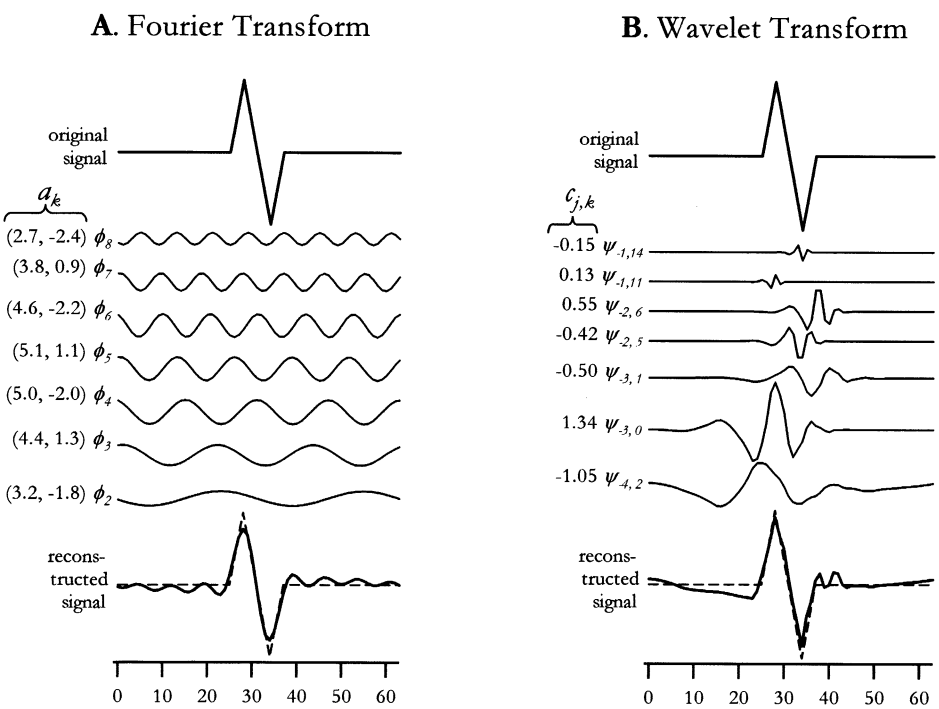



Fig. 1. Signal decomposition and reconstruction: Fourier versus Wavelet Transforms. (A) A 64-point bi-triangular profile (original signal, top trace) was created, its Fast Fourier Transform was computed, and the seven highest magnitude spectral coefficients were selected ($a_2$ to $a_8$ of Eq. (1)). The corresponding terms of the Fourier decomposition series are plotted below the original signal (middle traces). Numbers in brackets at the left of each wave name are the complex Fourier coefficients $a_k$, expressed in polar coordinates (magnitude, phase). The point-to-point sum of these seven terms produces a good approximation (solid line) of the original signal (dashed line). Abscissa values are in points. (B) The same original signal was decomposed by discrete wavelet transform. Here, seven terms of the series (Eq. (5)) were picked by the amplitude of their (real-valued) $c_{j,k}$ coefficients. The decomposing wavelets $\psi_{j,k}$, scaled and shifted according to Eq. (6), are plotted below the original signal with their corresponding $c_{j,k}$ coefficients. Note how the lower-frequency components of the series (*scaling index* $j = -4, -3$) account for the general profile of the signal, while the higher-frequency components ($j = -2, -1$) contribute local details. Every decomposition term carries information locally restricted to a time interval that depends on *shifting index* $k$, whereas the Fourier series' terms contribute to the whole extension of the signal. This reflects the compact support that characterizes the wavelet basis functions, versus the infinite support of Fourier basis functions, sine and cosine. The signal was again reconstructed by adding the chosen terms of the decomposition series. Wavelet decomposition captures the spiky nature of the signal more efficiently than Fourier analysis. This is due to the similarity between the signal's temporal profile and the wavelet (Daubechies-8) chosen for DWT analysis.

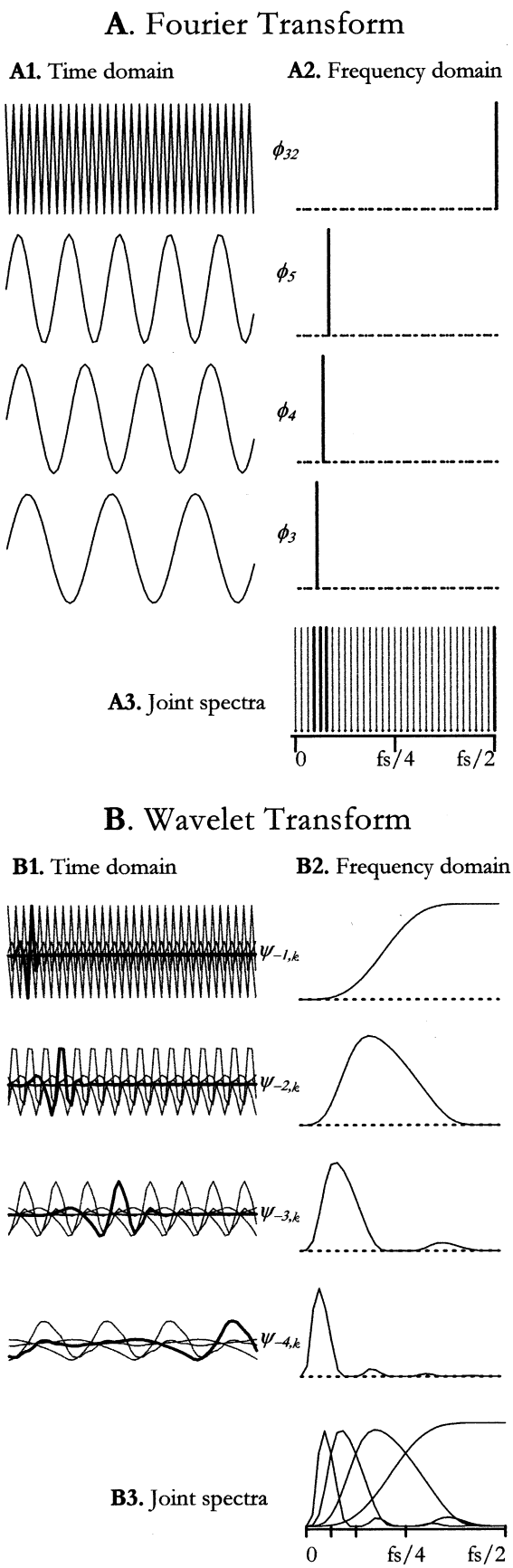



Fig. 2. (*Continued*)

## 2. Methods

### *2.1. Mathematical foundations of wavelet transform (WT)*

Fourier analysis states that any periodic function can be represented as an infinite, enumerable sum of trigonometric functions (Papoulis, 1962; Press et al., 1993) (Fig. 1A). In the complex representation of the Fourier Series, the fundamental equations are:

$$f(t) = \sum_k a_k \, \phi_k(t) \tag{1}$$

$$\phi_k(t) = \Phi(k\,t) = e^{i\,k\,w_0\,t} \tag{2}$$

$$\Phi\,(t) = e^{i\,w_0\,t} \tag{3}$$

Through dilations of the generating function $\Phi$, Eq. (2) generates a set of orthonormal functions $\phi_k$ that define a basis of the space of periodic functions (with period $2\pi/\omega_0$). The dilation index $k$ sets the specific frequency of the corresponding $\phi_k$. Factors $a_k \in \mathbf{C}$ are calculated by the inner product (Papoulis, 1962; Burrus et al., 1997)

$$a_k = \langle \phi_k(t), f(t) \rangle = \int_{\mathbf{R}} \phi_k^*(t) f(t)\, \mathrm{d}t, \tag{4}$$

where $\phi^*$ denotes the complex conjugate of function $\phi$.

Hence, the Fourier series maps a time-domain function into a pure frequency domain. Although no information is lost, the time-localization of specific features ceases to be explicitly available; it is hidden and distributed in the phase of the $a_k$. It should be noted that functions $\phi_k$ can also be interpreted as ideally narrow

Fig. 2. Time and frequency localization: Fourier versus Wavelet Transform. (A) FFT uses an orthonormal base of $n/2+1$ harmonic functions (sine and cosine) that span the entire signal bandwidth of $[0, f_s/2]$ at constant intervals of $f_s/n$, where $n$ is the number of points of the input vector sampled at $f_s$. The phase information of the complex Fourier coefficients $a_k$ shifts the corresponding base function $\phi_k$, but as they have infinite support, shifting does not carry any implicit time-localization. Thus, every Fourier basis function $\phi_k$ quantifies the energy content in a narrow frequency band of constant resolution $f_s/n$, but carries no information whatsoever about time-localization. The figure shows, for $n=64$, (A1) the functions $\phi_k$ for $k=3$, 4, 5 and 32, (A2) their corresponding spectra, and (A3) how the signal bandwidth is segmented by the 33 maximally narrow band-pass filter functions $\phi_k$. Dark lines indicate the specific $k$ values shown in A1 and A2. (B) Wavelets are *band-pass filters*, whose higher cut-off frequency is (almost exactly) twice the lower cut-off frequency. The higher the frequencies it lets pass, the broader the band. For $n=64$, (B1) shows all Daubechies-8 wavelets for $j=-1$ to $-4$, that is, wavelets $\psi_{-1,k}$ ($k=0$–31), $\psi_{-2,k}$ ($k=0$–15), $\psi_{-3,k}$ ($k=0$–7) and $\psi_{-4,k}$ ($k=0$–3). In every $j$-level, all functions differ only by translation (index $k$), and the wavelet corresponding to $k=0$ has been indicated by a darker line. (B2) shows the corresponding spectra for every $j$-level (the spectra are independent from index $k$). Note how the time-resolution is inversely proportional to the frequency-resolution (compare with A2). (B3) shows the four spectra found in B2 and how they complement each other to cover the signal bandwidth (spectra for $j<-4$ not shown).

band-pass filters that extract the energy content of the signal at a specific, infinitely narrow frequency band (Fig. 2A).

Wavelet analysis tackles the time-localization problem of signal features by replacing the infinite sine or cosine basis functions with compactly supported functions (i.e. functions that have most of their energy localized in a finite time interval) (Fig. 1B). This time-localization ability comes at the price of poorer frequency-resolution than in Fourier analysis (uncertainty principle) (Fig. 2B) (Papoulis, 1962; Meyer, 1993; Chui, 1997). Wavelet analysis can be applied to any finite-energy function $f \in L^2$ and maps $f$ from the time domain into a compound time–frequency space. The discrete wavelet transform represents function $f$ as an enumerable, linear combination of functions $\psi_{j,k}$:

$$f(t) = \sum_{j,k} c_{j,k}\, \psi_{j,k}(t) \tag{5}$$

Analogous to Eq. (2), functions $\psi_{j,k}$ (called *wavelets*) constitute an infinite, orthonormal basis of the space of finite-energy functions $L^2$, and are derived from a *mother wavelet* $\Psi$ as follows:

$$\psi_{j,k}(t) = \frac{1}{2^{j/2}}\, \Psi(2^j t - k) \tag{6}$$

Index $j$ (called the *scaling index*) changes the behavior of $\psi_{j,k}$ in frequency space, while index $k$ (called the *translation index*) shifts the wavelet along the time axis (Fig. 1B and Fig. 2B). Analogous to Eq. (4), coefficients $c_{j,k}$ are computed as:

$$c_{j,k} = \langle \psi_{j,k}(t), f(t) \rangle \tag{7}$$

There is, though, no unique equivalent to Eq. (3). Rather wavelet analysis provides several choices for the mother wavelet $\Psi$, which must comply with certain general requisites (like smoothness, compact support, zero mean, etc.), plus specific requisites depending on the kind of algorithm to be used (Meyer, 1993; Burrus et al., 1997; Chui, 1997). Depending on the application, there are two main kinds of wavelet algorithms: the continuous and the discrete forms. The term 'continuous wavelet transform' (CWT) is indistinctly used for the integral and the semidiscrete wavelet transform described in the literature (Meyer, 1993), and is generally implemented by analytical or numeric convolution. The discrete wavelet transform (DWT) uses orthonormal or bi-orthogonal wavelets and is most frequently computed by recursive multiplication of the data with particular filter matrices (Strang, 1993; Chui, 1997). The continuous versus discrete forms impose additional conditions upon $\Psi$. CWT demands, for example, that $\Psi$ is an analytical function. On the other hand, a DWT mother wavelet does not have to be given in a closed form, but it must generate through Eq. (6) an orthogonal, dyadic, non-redundant mapping onto time–frequency space (Meyer, 1993). As we aim for a fast spike sorting procedure, the WSC algorithm will use a DWT based on a length-8 Daubechies system and computed by a fast pyramidal algorithm (Press et al., 1993; Burrus et al., 1997; Chui, 1997).

### 2.2. Discrete wavelet transform using a pyramidal filter bank algorithm

The most frequently used method to calculate the DWT of an $n$-point data vector $\mathbf{v}$ is to implement a bank of quadrature mirror filters (QMF) and obtain an $n$-component output vector $\mathbf{c} = \mathrm{DWT}(\mathbf{v})$ (Burrus et al., 1997). The following pyramidal DWT algorithm is based on the Filter Bank implementation found in Press et al. (1993).

This algorithm demands $\mathbf{v}$ to be an evenly sampled data vector of $n = 2^p$ points, where $p$ has to be an integer $> 2$. The data vector $\mathbf{v}$ can be expressed in terms of the analog signal $f(t)$, the initial time $t_0$ and the sampling frequency $f_s$ as $\{v_i\}_{i=0}^{n-1} = \{f(t_0 + i/f_s)\}_{i=0}^{n-1}$. The sampling theorem (Press et al., 1993) asserts that, with a sampling frequency $f_s$, only a signal with bandwidth $[0, f_s/2]$ or less can be properly acquired. The pyramidal algorithm segments the frequency interval $[0, f_s/2]$ in $\log_2(n) - 1$ successive octaves, resulting in the dyadic time–frequency decomposition characteristic of the DWT.

Step one of the filter bank algorithm is to multiply data vector $\mathbf{v} = \{v_i\}_{i=0}^{n-1}$ by two rectangular matrices $\mathbf{A_1}$ and $\mathbf{B_1}$ that represent high- and low-pass filters, respectively. These two matrices perform a convolution upon the input and are defined as:and

$$\mathbf{A_1} = \overbrace{\begin{bmatrix} a_0 & a_1 & a_2 & \cdots & \cdots & \cdots & a_7 & & & & & \\ & & a_0 & a_1 & a_2 & \cdots & \cdots & \cdots & a_7 & & & \\ & & & & a_0 & a_1 & a_2 & \cdots & \cdots & \cdots & a_7 & \\ & & & & & \ddots & \ddots & & & & & \ddots \\ a_6 & a_7 & & & & & a_0 & a_1 & a_2 & \cdots & \cdots & a_5 \\ a_4 & \cdots & a_6 & a_7 & & & & & a_0 & a_1 & a_2 & a_3 \\ a_2 & \cdots & \cdots & \cdots & a_6 & a_7 & & & & & a_0 & a_1 \end{bmatrix}}^{n \text{ columns}} \begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix} \overbrace{\begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix}}^{(n/2)\ rows} \tag{8}$$

$$\mathbf{B_1} = \overbrace{\begin{bmatrix} b_0 & b_1 & b_2 & \cdots & \cdots & \cdots & b_7 & & & & & \\ & & b_0 & b_1 & b_2 & \cdots & \cdots & \cdots & b_7 & & & \\ & & & & b_0 & b_1 & b_2 & \cdots & \cdots & \cdots & b_7 & \\ & & & & & \ddots & \ddots & & & & & \ddots \\ b_6 & b_7 & & & & & b_0 & b_1 & b_2 & \cdots & \cdots & b_5 \\ b_4 & \cdots & b_6 & b_7 & & & & & b_0 & b_1 & b_2 & b_3 \\ b_2 & \cdots & \cdots & \cdots & b_6 & b_7 & & & & & b_0 & b_1 \end{bmatrix}}^{n \text{ columns}} \begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix} \overbrace{\begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix}}^{(n/2)\ rows} \tag{9}$$

Table 1
Daubechies-8 filter coefficients[a]

| $i$ | $a_i$ | $b_i$ |
|---|---|---|
| 0 | 0.23037781330890 | 0.01059740178507 |
| 1 | 0.71484657055292 | 0.03288301166689 |
| 2 | 0.63088076792986 | −0.03084138183556 |
| 3 | −0.02798376941686 | −0.18703481171909 |
| 4 | −0.18703481171909 | 0.02798376941686 |
| 5 | 0.03084138183556 | 0.63088076792986 |
| 6 | 0.03288301166689 | −0.71484657055292 |
| 7 | −0.01059740178507 | 0.23037781330890 |

[a] Source: Burrus et al. (1997), p. 79.

The filter matrices $\mathbf{A_1}$ and $\mathbf{B_1}$ are constructed by kernels $\{a_i\}_{i=0}^{7}$ and $\{b_i\}_{i=0}^{7}$ that are associated with the chosen mother wavelet (in this case, Daubechies-8, hence eight filter coefficients) (Table 1) (Press et al., 1993; Burrus et al., 1997; Chui, 1997). Each row of the matrix contains the eight non-zero numbers of the respective filter kernel, *right-shifted two positions* with respect to the upper row. All other $n-8$ positions contain zeroes. To tackle the unavoidable edge effect associated with finite numeric convolution, we used circulant, wrap-around filter matrices (see the triangular non-zero zone in the lower left corner of matrices $\mathbf{A_1}$ and $\mathbf{B_1}$) that treat the input vector as a periodic signal. This results in a circulant, rectangular matrix (number of rows = half number of columns) that produces a halving or decimation (*downsampling* by two) of the input vector.

The *filtering* action of the matrices is given by the respective filter kernel. Kernel $\{a_i\}$ is a high-pass filtering kernel when convoluted with the signal, while kernel $\{b_i\}$ acts as a low-pass filter. Hence, the rectangular, circulant matrices $\mathbf{A_1}$ and $\mathbf{B_1}$ simultaneously downsample and filter (high- and low-pass filters, respectively) the input vector. The couple $\mathbf{A_1}$ and $\mathbf{B_1}$ implement the first step of the QMF bank. The general characteristics of quadrature mirror filters' behavior are shown in Fig. 3.

Matrices $\mathbf{A_1}$ and $\mathbf{B_1}$ are applied to the input vector $\mathbf{v}$, resulting in two output vectors, $\mathbf{d}_{-1}$ and $\mathbf{r}_{-1}$ (Fig. 4).

$$\mathbf{A_1 v} = \mathbf{d}_{-1} \text{ and } \mathbf{B_1 v} = \mathbf{r}_{-1} \tag{10}$$

Due to the downsampling characteristic of $\mathbf{A_1}$ and $\mathbf{B_1}$, output vectors $\mathbf{d}_{-1}$ and $\mathbf{r}_{-1}$ both have $n/2$ data points. Vector $\mathbf{d}_{-1}$ is the output of the high-pass filter and contains *detail* information of the input vector $\mathbf{v}$ within the approximate frequency range $[f_s/4, f_s/2]$ (Fig. 2B2 and Fig. 3). Vector $\mathbf{r}_{-1}$ is the output of the low-pass filter and contains the *rough* information of the input vector $\mathbf{v}$ within the approximate frequency range $[0, f_s/4]$. The subscript $-1$ reflects the fact that $\mathbf{d}_{-1}$ and $\mathbf{r}_{-1}$ were obtained in the first step of the DWT process. Vector $\mathbf{d}_{-1}$ contains exactly the $c_{j,k}$ coefficients of Eqs. (5) and (7) for $j=-1$. The negative sign of $j$ is due to a common convention in the field of wavelet analysis and indicates that the output's frequency content is less than the input bandwidth. A more positive index $j$ stands for higher frequency (Chui, 1997).

The $n/2$-points vector $\mathbf{r}_{-1}$ enters the second step of the algorithm, while the $n/2$-points vector $\mathbf{d}_{-1}$ forms part of the global output vector $\mathbf{c}$

$(\mathbf{d}_{-1} = \{c_{-1,k}\}_{k=0}^{\frac{n}{2}-1})$ (Fig. 4).

Step two of the filter bank algorithm begins with the construction of matrices $\mathbf{A_2}$ and $\mathbf{B_2}$, both with $n/4$ rows and $n/2$ columns. Replacing $n$ with $n/2$ in 8 9, we obtain the rectangular, circulant matrices $\mathbf{A_2}$ and $\mathbf{B_2}$. These are applied to $\mathbf{r}_{-1}$ to obtain step 2 output vectors $\mathbf{d}_{-2}$ and $\mathbf{r}_{-2}$.

$$\mathbf{A_2 r}_{-1} = \mathbf{d}_{-2} \text{ and } \mathbf{B_2 r}_{-1} = \mathbf{r}_{-2} \tag{11}$$

Vectors $\mathbf{d}_{-2}$ and $\mathbf{r}_{-2}$ have $n/4$ points each (half the number of $\mathbf{r}_{-1}$) and their approximate frequency contents are $[f_s/8, f_s/4]$ and $[0, f_s/8]$, respectively. Again $\mathbf{d}_{-2}$ contains exactly the $c_{j,k}$ coefficients for $j=-2$ and forms part of the global output vector $\mathbf{c}$

$(\mathbf{d}_{-2} = \{c_{-2,k}\}_{k=0}^{\frac{n}{4}-1})$,

whereas $\mathbf{r}_{-2}$ will be used as input for step 3.

In general, step $m$ of the filter bank algorithm will be

$$\mathbf{A}_m\, \mathbf{r}_{-(m-1)} = \mathbf{d}_{-m} \text{ and } \mathbf{B}_m\, \mathbf{r}_{-(m-1)} = \mathbf{r}_{-m}. \tag{12}$$

Note that step $m$ of the algorithm calculates the $c_{j,k}$ coefficients for $j=-m$. Matrices $\mathbf{A}_m$ and $\mathbf{B}_m$ have $n/2^m$ rows and $n/2^{m-1}$ columns. Output vectors $\mathbf{d}_{-m}$ and $\mathbf{r}_{-m}$ have $n/2^m$ points each and their approximate frequency contents are $[f_s/2^{m+1}, f_s/2^m]$ and $[0, f_s/2^{m+1}]$, respectively.

The recursive procedure stops for the index $M = \log_2(n/2)$ that produces vectors $\mathbf{d}_{-M}$ and $\mathbf{r}_{-M}$ of length 2. The final result of the DWT of input vector $\mathbf{v}$ is an $n$-component vector that is assembled with the coefficients of all $n-2$ detail vectors $(\mathbf{d}_{-M}, \mathbf{d}_{-(M-1)}, \ldots, \mathbf{d}_{-1})$, and the residual rough feature vector $\mathbf{r}_{-M}$ (Fig. 4).

From a computational standpoint, the dual filtering process embodied by matrices $\mathbf{A}_m$, $\mathbf{B}_m$, $m=1, 2, \ldots, M$ acting as a filter bank is extremely efficient. For eight filter coefficients, only $8/n$ matrix entries are non-zero, thus a very fast numerical algorithm can compute the matrix multiplications. In fact, though the DWT and Fast Fourier Transform both make use of matrix factorization, the DWT is faster (Strang, 1993).

In summary the DWT calculates a 'wavelet spectrum', consisting of $n$ $c_{j,k}$ coefficients, from each temporal profile of $n$ points.

### 2.3. Spike sorting based on DWT coefficients: the WSC method

As for all spike-sorting algorithms, the initial step prior to the WSC method is to extract the events that could represent spikes from the multi-unitary recording. Many schemes, such as a simple peak detection algorithm, can be used. Once an event has been detected, a segment of $n = 2^p$ points (32, 64, 128, etc.) is selected straddling the peak. If $N_s$ spike events are detected, a family of $N_s$ temporal profiles $\{\mathbf{v^i}\}_{i=1}^{N_s}$ is created. For $n = 64$ data points, each profile $\mathbf{v^i}$ can be written as $\mathbf{v^i} = \{v_0^i, v_1^i, \ldots, v_{63}^i\}$. The next and central step is to apply the procedure explained in Section 2.2 to obtain $\mathbf{c^i}$, the DWT vector associated with profile $\mathbf{v^i}$. For $n = 64$, $\mathbf{c^i}$ can be written as $\mathbf{c^i} = \{r^i_{-5,0}, r^i_{-5,1}, d^i_{-5,0}, d^i_{-5,1}, d^i_{-4,0}, \ldots, d^i_{-1,30}, d^i_{-1,31}\}$.

The third step is to evaluate the relative variability of coefficients $c_{j,k}$ across the population of DWT vectors $\mathbf{c}^i$ in order to identify the coefficients that exhibit the greatest potential to differentiate spikes. Variability can be assessed with many techniques. A simple method is to calculate the average $\bar{c}_{j,k}$ and S.D. $\sigma_{j,k}$ for each coefficient $c_{j,k}$ and use them to assess differentiating potential:

$$\bar{c}_{j,k} = \frac{1}{N_s}\sum_{i=1}^{N_s} c^i_{j,k}, \quad \sigma_{j,k} = \sqrt{\frac{1}{(N_s - 1)}\sum_{i=1}^{N_s}(c^i_{j,k} - \bar{c}_{j,k})^2}$$

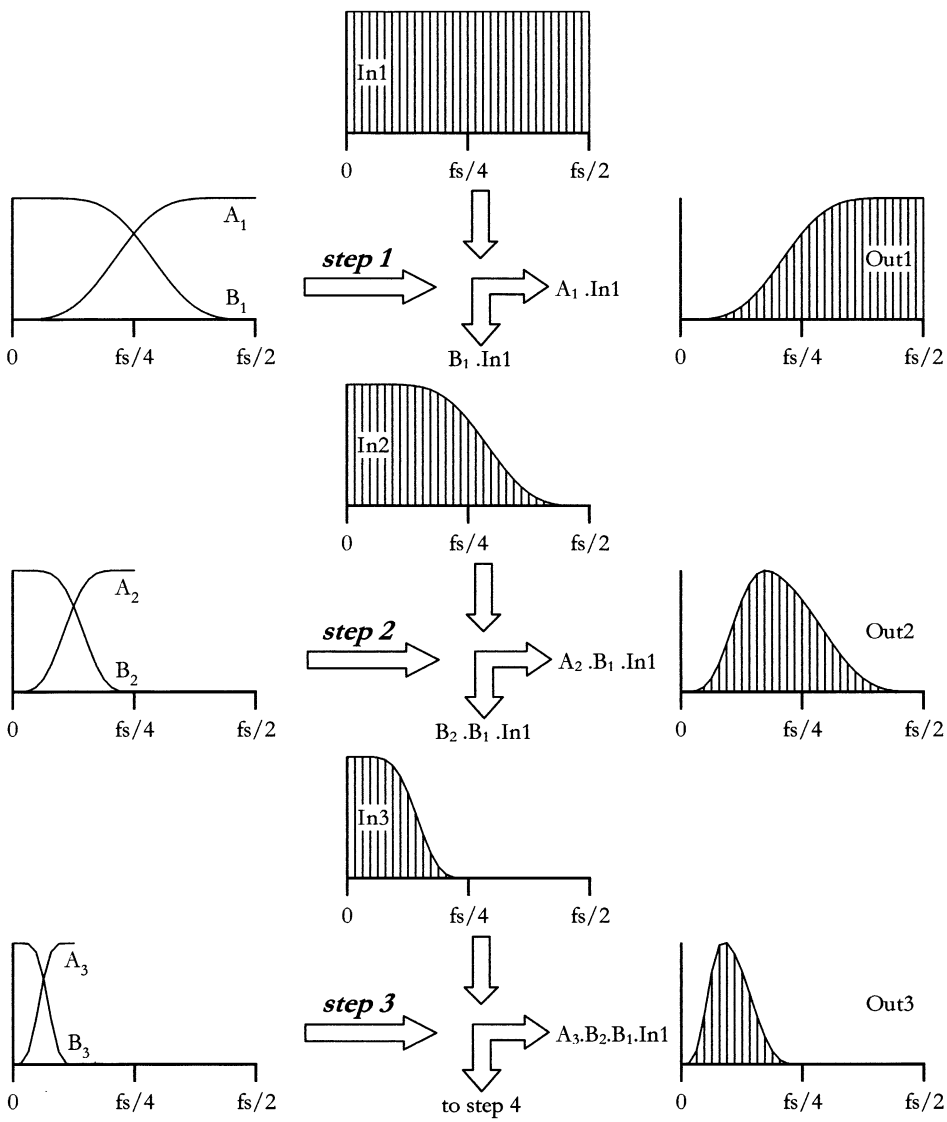
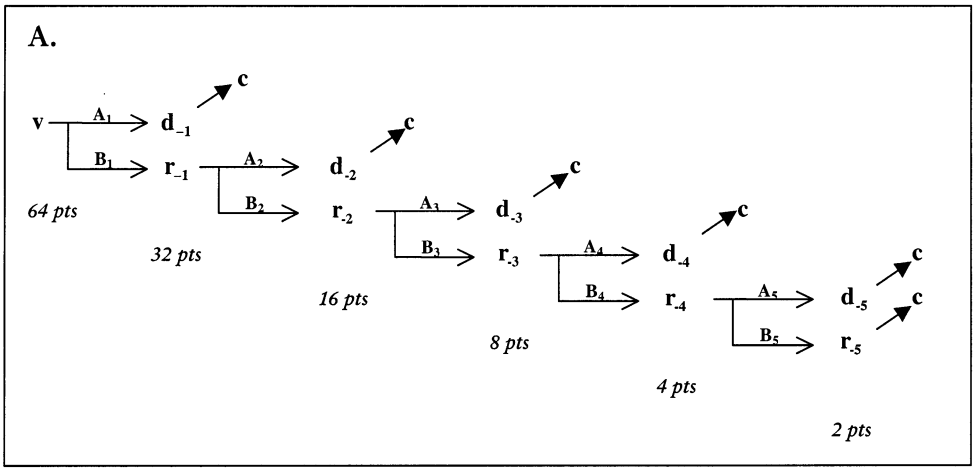



Fig. 3. Operation of a quadrature mirror filter bank in frequency space. The first three steps of the pyramidal algorithm are shown in frequency space. For demonstration purposes, the initial input signal (In1) has a flat spectrum in the range $[0, f_s/2]$. Step 1 takes In1, applies filters $\mathbf{A_1}$ and $\mathbf{B_1}$, which are symmetric around midfrequency (*mirror* feature of QMFs), and produces two signals whose spectra are complementary to each other (*quadrature* feature of QMFs). The output containing the higher frequencies is the global output of step 1 (Out1). The other output, containing the lower frequencies, serves as input for step 2 (In2). Filters $\mathbf{A_2}$ and $\mathbf{B_2}$ inherit the same QMF characteristics, but are compressed in half the frequency range. Notice the equivalence of the output spectra (Out1, Out2, Out3) with the wavelet spectra shown in Fig. 2B.




**B.**

| | | | | | | |
|---|---|---|---|---|---|---|
| Vector **c** components | = | $\{\mathbf{r}_{-5}$, | $\mathbf{d}_{-5}$, | $\mathbf{d}_{-4}$, | $\mathbf{d}_{-3}$, | $\mathbf{d}_{-2}$, | $\mathbf{d}_{-1}\}$ |
| Explicit coefficients | = | $\{\{r_{-5,0}, r_{-5,1}\}$ | $\{c_{-5,k}\}_{k=0}^{1}$ | $\{c_{-4,k}\}_{k=0}^{3}$ | $\{c_{-3,k}\}_{k=0}^{7}$ | $\{c_{-2,k}\}_{k=0}^{15}$ | $\{c_{-1,k}\}_{k=0}^{31}\}$ |
| Approximate frequency decomposition | = | $\left[0, \frac{f_s}{64}\right]$ | $\left[\frac{f_s}{64}, \frac{f_s}{32}\right]$ | $\left[\frac{f_s}{32}, \frac{f_s}{16}\right]$ | $\left[\frac{f_s}{16}, \frac{f_s}{8}\right]$ | $\left[\frac{f_s}{8}, \frac{f_s}{4}\right]$ | $\left[\frac{f_s}{4}, \frac{f_s}{2}\right]$ |

Fig. 4. Schematic operation of the pyramidal DWT algorithm for a 64-point vector **v**. (A) The initial step applies matrices $\mathbf{A_1}$ and $\mathbf{B_1}$ upon **v**. $\mathbf{A_1 v}$ produces vector $\mathbf{d}_{-1}$ (32 points), which is collected into the global output vector **c** (arrow), while $\mathbf{B_1 v}$ produces $\mathbf{r}_{-1}$ (32 points). Step 2 takes vector $\mathbf{r}_{-1}$ and applies matrices $\mathbf{A_2}$ and $\mathbf{B_2}$, generating 16-point vectors $\mathbf{d}_{-2}$, collected into global output **c** (arrow), and $\mathbf{r}_{-2}$, which enters step 3. The iteration stops after step 5, with two-point output vectors $\mathbf{d}_{-5}$ and $\mathbf{r}_{-5}$. The output vector $\mathbf{c} = \mathrm{DWT}(\mathbf{v})$ is comprised of the following elements: $\mathbf{r}_{-5}$, $\mathbf{d}_{-5}$, $\mathbf{d}_{-4}$, $\mathbf{d}_{-3}$, $\mathbf{d}_{-2}$ and $\mathbf{d}_{-1}$. (B) Relationship between components of output vector **c** and the decomposition of frequency space.

The criteria for differentiating potential are: large S.D., large average and, most fundamentally, a bimodal or multimodal distribution. All $c_{j,k}$ are plotted and the $m$ $c_{j,k}$ coefficients with highest differentiating potential are visually selected. These $m$ coefficients capture most of the variability and define the $m$-dimensional space $S_m$ mentioned in Section 1.

$$\overbrace{\mathbf{v}^{\mathbf{i}} = \{v_0, v_1, \ldots\ldots, v_{63}\}}^{\text{Temporal profile}} \rightarrow \overbrace{\mathbf{c}^{\mathbf{i}} = \{c^i_{-5,0}, c^i_{-5,1}, c^i_{-4,0}, \ldots\ldots, c^i_{-1,30}, c^i_{-1,31}\}}^{\text{Wavelet coefficients}} \tag{13}$$

$$\rightarrow \mathbf{c}^{\boldsymbol{i}}_{\boldsymbol{m}} = \overbrace{\{c^i_{-j_1,k_1}, c^i_{-j_2,k_2}, \ldots\ldots, c^i_{-j_m,k_m}\}}^{m \text{ coefficients as separating parameters}} \tag{14}$$

To separate the different types of profiles all $m(m-1)/2$ possible 2-D projections $c^i_{j',k'}$ versus $c^i_{j'',k''}$ are plotted. The set of plots is visually scanned to find the graph (or graphs) that exhibit clear clusters of points.

The final step of the WSC method is built around a graphic software sorter interface that allows the user to visually scan the set of projections and to manually delimit clusters.

## 3. Application of WSC method on artificial data

The WSC method was tested on a specially designed artificial spike train based on three spike templates and an important noise component. To realistically mimic the characteristics of the noise found in extracellular recordings, we extracted 0.2 ms (4096 points sampled at 20 kHz) of real noise from data obtained in the petrous ganglion of a cat. The noise phase and amplitude spectra were analyzed. The magnitude information indicated a typical $1/f$ noise signature, while phases were, as expected, evenly distributed (Fig. 5B). Artificial noise, with the same phase and magnitude distribution of the sampled noise, was generated by inverse Fourier transform and divided by its rms value. The resulting zero-mean 'pure-noise' signal contained $2^{16}$ data points,

lasted 3.2 s, and had unitary rms and near unitary S.D.

To assess the sorting capabilities of the WSC method, we made two of the three spike templates (TII and TIII) extremely similar, with a common triphasic time course and comparable peak amplitudes (Fig. 5A). Template TI, on the other hand, was biphasic and had a larger peak amplitude. The amplitude of the templates was expressed in terms of the noise rms value.

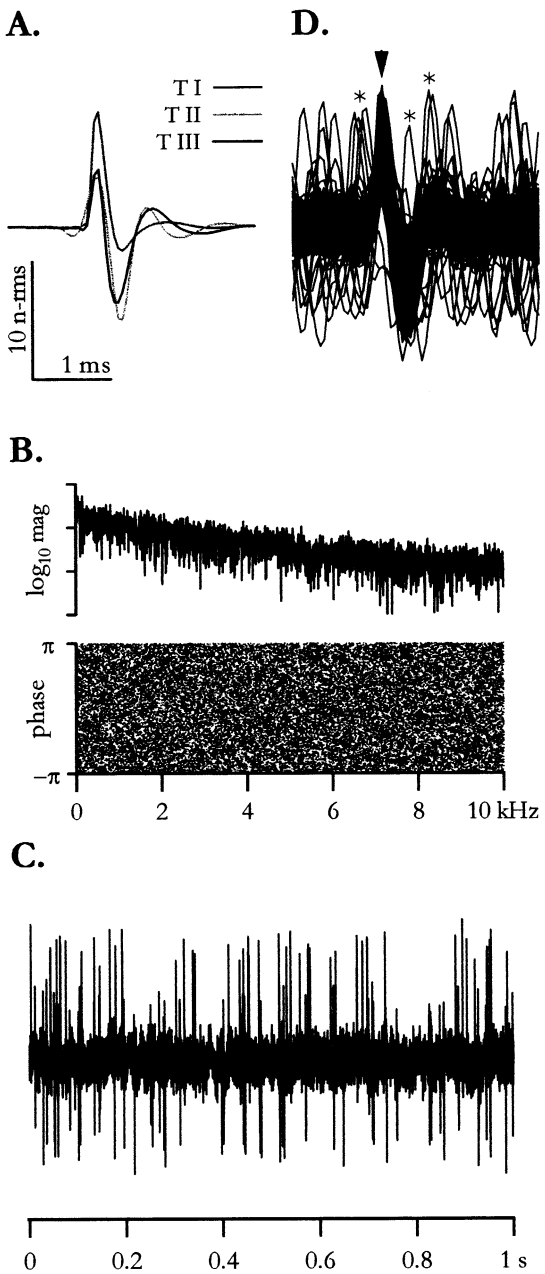



Fig. 5. Artificial spike train construction. (A) Sixty-four-point artificial spike templates. Template TI differs substantially from templates TII and TIII. Amplitudes are expressed in terms of the noise's rms value (n-rms, see text). (B) Magnitude (upper trace) and phase (lower trace) spectra of artificial noise. The magnitude approximately follows a $1/f$ behavior between 2 and 10 kHz, visible in a log/log plot (not shown). (C) First 1-s interval of artificial spike train. (D). Superposition of all 300 spikes extracted from the artificial spike train in C. Same time and amplitude scaling as in (A). Spikes were centered at their positive peak, at point number 24 (arrowhead). Peaks at other time locations within this plot are due to superposition (*).

One hundred instances of each template were randomly inserted by point-to-point addition of the template to the noise background. The *insertion point* was defined as the point in the noise signal where the template peak was added (Fig. 5C,D). The resulting spike train mimicked three neurons firing independently at an average rate of 30 spikes/s and contained 19 superpositions with a delay of less than 1 ms between peaks (Fig. 5C).

The information about the insertion points served to construct a 'near-to-perfect spike detector'. Around each insertion point the local maximum (peak) was found and a 64-point segment bracketing that peak was extracted (23 points before, 40 points after) to produce a spike profile (Fig. 5D). Because of our perfect knowledge of the insertion points, superposition events always generated separate profiles, and no spikes went undetected. Yet the detected spike was centered at its local maximum, thus introducing a temporal jitter of up to two points (0.1 ms) around the actual insertion point. This temporal jitter, caused by high-amplitude noise, is typical for most spike-detection algorithms.

The DWT was computed for every spike profile (Fig. 6A,B) and all 300 DWT vectors were plotted along with their average and S.D. (Fig. 6D). The plot was visually scanned to select the $c_{j,k}$ coefficients that exhibited the greatest potential to differentiate spikes, as described in Section 2.3 (Fig. 6C,D). For this specific artificial spike train, four such components were finally selected ($c_{-5,1}$, $c_{-4,2}$, $c_{-3,7}$ and $c_{-2,3}$). Our graphic software sorter interface presented all six possible 2-D projections of $S_4$ to allow the user to manually delimit each distinguishable cluster (Fig. 7).

Five untrained observers were briefed in the method and asked to use the graphic interface to construct clusters. The following Classification Matrix quantified their performance:

*Observed Classification Matrix*

$$= \begin{bmatrix} & \text{Type I} & \text{Type II} & \text{Type III} \\ \text{Class 1} & d_1 & r_4 & r_5 \\ \text{Class 2} & r_1 & d_2 & r_6 \\ \text{Class 3} & r_2 & r_3 & d_3 \end{bmatrix} \quad (15)$$

The *Type* (I, II, III) represents the spikes belonging to each template, while the *Class* (1, 2 or 3) represents the clusters distinguished by the user. A perfect performance of the user and the sorting procedure, at 100 spikes per type, should produce the following Classification Matrix:

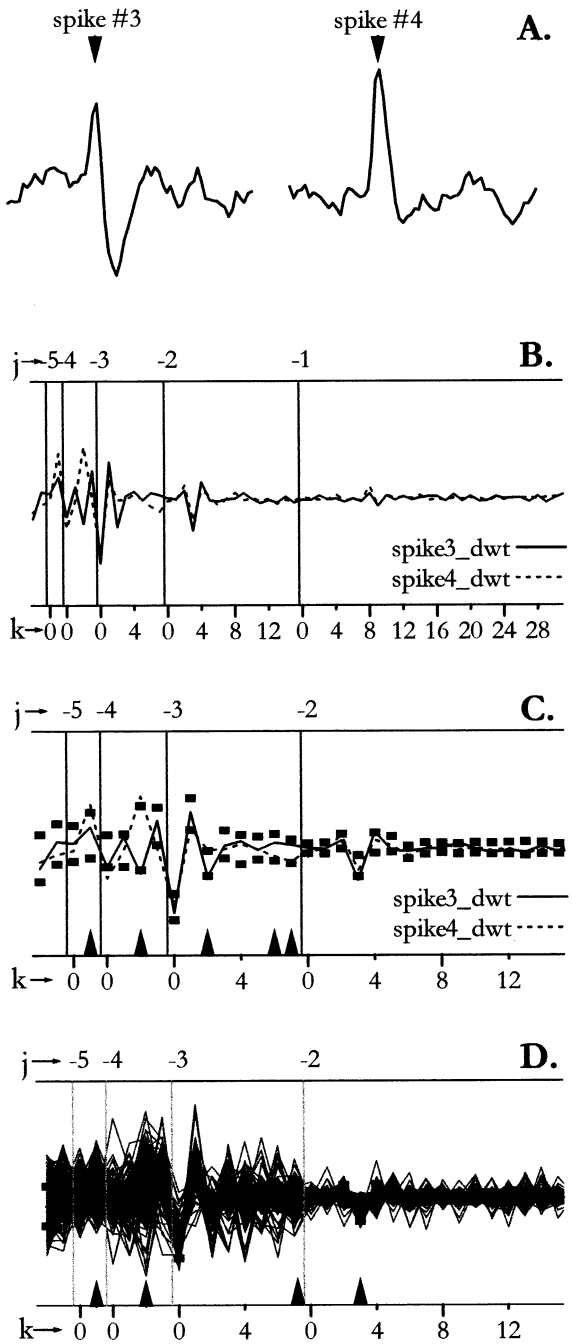



Fig. 6.

*Expected Classification Matrix*

$$= \begin{bmatrix} & \text{Type I} & \text{Type II} & \text{Type III} \\ \text{Class 1} & 100 & 0 & 0 \\ \text{Class 2} & 0 & 100 & 0 \\ \text{Class 3} & 0 & 0 & 100 \end{bmatrix} \quad (16)$$

For every observer, we quantified *misclassified* ($\Sigma r_i$) and *unclassified* profiles ($N_s - (\Sigma r_i + \Sigma d_i)$). To quantify the similitude between the observer Classification Matrix (Eq. (15)) and the expected result (Eq. (16)), an *Error Index* was defined as the root mean square difference between expected and observed values:

$$\textit{Error Index} = \sqrt{\frac{\sum (d_i - 100)^2 + \sum r_k^2}{\sum d_i + \sum r_k}} \quad (17)$$

A lower Error Index indicates less misclassified and unclassified events, and therefore a better spike-sorting performance.

### 3.1. Assessment of the WSC performance

Table 2 shows the observed Classification Matrices for all five observers. All observers detected three classes, with comparable numbers per class between observers (85–110 spikes/class). The average number of unclassified events was 33.4 (S.D. = 19.2). Interestingly, all observers were able to separate types II and III into distinct classes. Correctly classified spikes ranged 79–88% for type I, 67–93% for type II and 76–79% for type III (diagonal values of the Classification Matrices). Classes 1 and 3 proved highly specific for types I and III, respectively (0–1 and 4–6% misclassifications, respectively). There were 10–16% misclassifications of type III into class 2, a rather encouraging result considering the extreme similarity between templates TII and TIII. Matrices of the first three observers were very similar to the results obtained by an expert in the WSC software (interface programmer), indicating that little expertise is needed to handle the WSC sorter.

To compare the relative merit of the WSC method with respect to previously published methods, sorting coefficients were calculated for RFS (Salganicoff et al., 1988; Sarna et al., 1988), and PCA (Eggermont et al., 1983). Two-dimensional projections of the $S_3$ space generated by the three major PCA coefficients (which together accounted for 71.4% of the variability), and of

Fig. 6. The WSC method. (A) The WSC method will be explained by following the data transformations of two spikes generated by different templates. (B) DWT vectors associated with both spikes. Note how the $c_{j,k}$ coefficients for $j = -1$, $k = 0,\ldots, 31$ have little differentiation potential, represent mostly noise (see discussion) and can therefore be discarded. (C) DWT vectors associated with both spikes, but restricted to $j = -5, -4, -3, -2$. Here, and in the subsequent plot, square markers reflect the variation of every $c_{j,k}$ coefficient over the entire spike population (average of the total population of spikes $\pm 1$ S.D.). Some $c_{j,k}$ coefficients differ greatly between the two vectors (arrowheads), enabling a separation of these two spike profiles. However, they do not guarantee an appropriate separation for the entire spike population. (D) The determination of which $c_{j,k}$ coefficients are useful to sort spikes in the population must be done by analyzing the complete population of DWT vectors. The plot shows all 300 DWT vectors (average: black trace). Coefficients with high differentiation potential are those with a bimodal or multimodal distribution (better visualized at higher magnifications), a large S.D. and/or a large amplitude. Arrowheads mark the four differentiating $c_{j,k}$ coefficients used for this particular spike train.

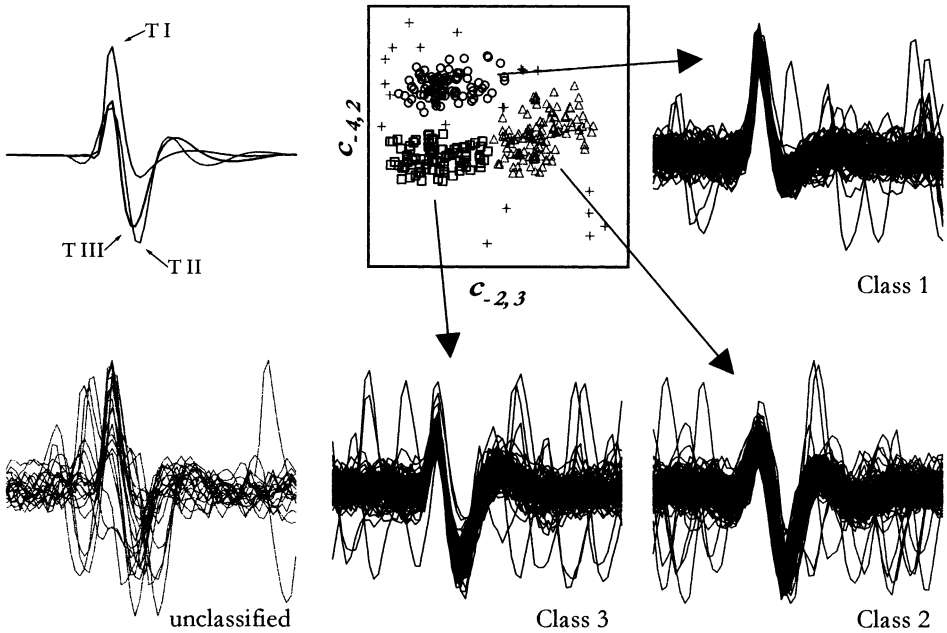



Fig. 7. Spike classification with WSC. The 2-D projection $c_{-4,2}$ versus $c_{-2,3}$ clearly separates the complete population into three classes that are quite specific for the three types TI–TIII. The set of unclassified events (+) contains mostly small-delay superpositions.

Table 2
Classification Matrices for WSC method (five observers)

| | Observers | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | JCL | | | GM | | | CC | | | JM | | | PH | | |
| | TI | TII | TIII | TI | TII | TIII | TI | TII | TIII | TI | TII | TIII | TI | TII | TIII |
| C 1 | 88 | 0 | 0 | 85 | 0 | 1 | 86 | 0 | 0 | 80 | 0 | 0 | 79 | 0 | 0 |
| C 2 | 1 | 90 | 15 | 2 | 90 | 16 | 1 | 93 | 16 | 0 | 80 | 16 | 0 | 67 | 10 |
| C 3 | 3 | 3 | 79 | 3 | 3 | 79 | 2 | 3 | 79 | 2 | 2 | 79 | 2 | 2 | 76 |

the $S_4$ space generated by four RFS coefficients were presented using the same graphic interface. All observers classified the 300 spike profiles with these two methods (Fig. 8). As with the WSC procedure, the efficiency of the PCA and RFS methods was assessed by tabulating the Classification Matrices and calculating the misclassified and unclassified profiles as well as the Error Index for each observer.

All five observers failed to separate spike types II and III using PCA coefficients, grouping them all into one single class (C2). Therefore, misclassification ranged near 100 spikes for PCA (Table 3). The RFS method had similar problems separating these two profiles, as 2/3 of type II spikes were misclassified as Class 3. All three methods generated comparable amounts of unclassified profiles.

The Error Index ranks the WSC method as best performer, followed by the RFS. The PCA method showed the worst performance in our artificial spike train.

## 4. Discussion

Wavelet analysis is ideally suited to separate coarse signal features from components with higher frequency contents. Unlike Short-Time Fourier Transform, the time-dependent frequency-filtering function is not multiplied by a separate, adjustable time-windowing function that equally applies for all frequencies. In wavelet analysis the frequency-filtering function acts as an oscillating time-window itself, whose width is pre-adjusted to the frequency band it lets pass (Figs. 1 and 2) (Meyer, 1993). Thus the wavelet transform of neurophysiological signals, especially when implemented with a DWT algorithm, non-redundantly separates *fast* from *slow transients*. High-frequency $c_{j,k}$ coefficients (more positive $j$) describe fast components (sharp edges and steep slopes) with a very accurate time-localization. Slow trends (repolarizations or baseline fluctuations) are represented by the low-frequency $c_{j,k}$ coefficients (more negative $j$), whose time-localization is accord-

ingly less accurate (Figs. 1 and 2). This characteristic of joint time–frequency localization is of great use in highly non-stationary, broad-spectrum signals typical in biomedical signal processing (Aldroubi and Unser, 1996; Samar et al., 1999). Wavelet analysis has introduced a common theoretical framework for diverse signal processing algorithms like sub-band coding, pyramidal algorithms and atomic decomposition (e.g. Gabor analysis) (Meyer, 1993; Sinkkonen et al., 1995). It has also made way for new techniques in fields like

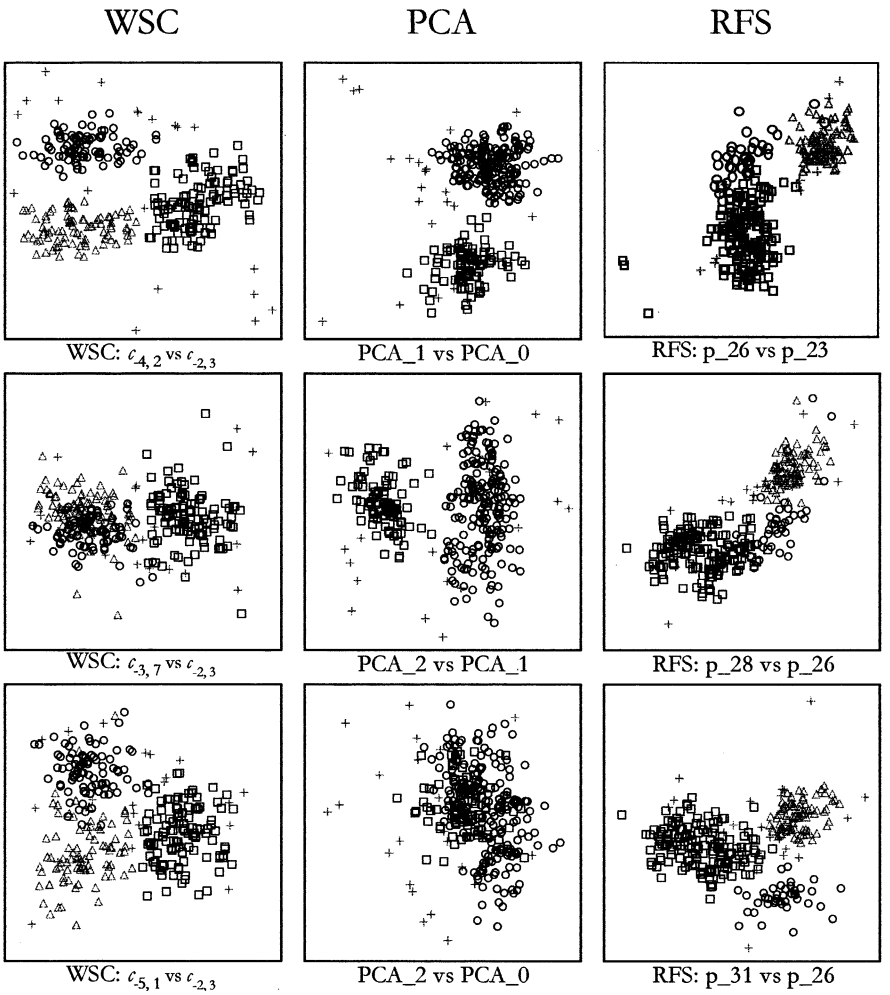



Fig. 8. Clustering with WSC, PCA and RFS: a comparison. Clustering achieved by observer CC (three 2-D projections per method shown). By construction, every cluster should contain 100 elements. The PCA method, using the three principal components (71.4% variability), only permitted the definition of two distinct clusters. One class (O) groups events belonging to templates TII and TIII. Even with six PCA coefficients (accounting for 77.6% variability), only these two spike classes were discriminated by all observers (see Section 4). The RFS method, using four coefficients, defines three clusters (lower graph), but one class is overrepresented (□) at the expense of another class (○). Unclassified spikes: (+).

Table 3
Comparison of classification methods

| Observer | Unclassified | | | Misclassified | | | Error index | | |
|---|---|---|---|---|---|---|---|---|---|
| | WSC | PCA | RFS | WSC | PCA | RFS | WSC | PCA | RFS |
| JCL | 21 | 8 | 9 | 22 | 99 | 61 | 30.5 | 140.8 | 89.2 |
| GM | 21 | 23 | 11 | 25 | 93 | 63 | 32.3 | 136.5 | 88.7 |
| CC | 20 | 22 | 22 | 22 | 92 | 61 | 30.9 | 135.6 | 86.7 |
| JM | 41 | 33 | 27 | 20 | 94 | 56 | 38.8 | 138.8 | 85.7 |
| PH | 64 | 26 | 38 | 14 | 92 | 58 | 47.1 | 136.5 | 92.8 |
| Avg. | 33.4 | 22.4 | 21.4 | 20.6 | 94.0 | 59.8 | 35.9 | 137.7 | 88.6 |
| S.D. | 19.2 | 9.1 | 11.9 | 4.1 | 2.9 | 2.8 | 7.1 | 2.1 | 2.8 |

advanced non-stationary noise filtering (Bertrand et al., 1994; Burrus et al., 1997; Zouridakis and Tam, 1997) or feature extraction (Trejo and Shensa, 1999).

Our WSC method uses the DWT coefficients derived from temporal spike-profiles as feature-extraction parameters. The central idea of the WSC method is that the same few features that make two spike profiles look different to the human eye will also differentiate them in the time–frequency wavelet spectrum. In other words, there has to be a reduced set of $c_{j,k}$ coefficients that can be used as elements for an unspecialized pattern-recognition algorithm.

The WSC method, as measured by the Error Index, outperformed the PCA and RFS methods. Spikes with extremely similar temporal profiles and in the masking presence of high-amplitude noise that could not be separated or were largely misclassified by these previous methods, were correctly classified by the WSC method. In particular, the PCA method completely failed to distinguish three classes.

The Principal Component Analysis returns the PCA coefficients in descending order of importance. Following the method's indications, we used the first three PCA coefficients, which accounted for 52.4, 14.6 and 4.4% of the variability, respectively, summing 71.4% of the variability. According to our five untrained observers, these three coefficients grouped the spikes originating from templates TII and TIII together into a single class.

To ensure a completely fair comparison between methods, we expanded the clustering process to an $S_6$ space by adding the next three PCA coefficients, of 3.4, 2.9 and 2.7% variability, respectively, thus accounting for a total of 80.3% of the variability. None of our observers distinguished three clusters in any of the 15 projections of the $S_6$ space. Two clusters were separated only in those projections that included one of the three major PCA coefficients and resulted in the separation of the same two classes obtained with the first three PCA coefficients. This indicates that the PCA method does not effectively distinguish between spikes generated by two very similar templates (TII and TIII), even when less important coefficients are included in the analysis.

Noise is a deleterious factor that hinders all spike-sorting methods (especially the RFS method). The WSC method is particularly noise resistant because it allows for the rejection of those wavelet coefficients that do not contain spike-related information. These include level $j = -1$ coefficients, which contain mainly high-frequency noise in the approximate frequency range 5–10 kHz, and the 'rough feature' coefficients $r_{-5,0}$ and $r_{-5,1}$ (Fig. 6B). Furthermore, those coefficients of level $j = -2,\ldots,-5$, whose index $k$ places them outside the spike duration, could also be algorithmically identified. By discarding all noise-related coefficients from classification, the WSC method performs non-stationary noise filtering and implicitly compresses the data, which can then be classified more accurately.

The choice of Daubechies-8 as the basic (or mother wavelet) for our analysis can appear rather arbitrary given the many mother wavelets available for QMF bank implementation (Battle-Lemarié, B-splines, Coiflet) (Chui, 1997). But two powerful reasons justify this choice. First, as the basic idea of the WSC method is to capture in the smallest possible number of coefficients the distinctive features of the spikes, it is convenient to use spike-shaped wavelets for analysis, as is the case of Daubechies wavelets (Fig. 1). Thus we can expect the inner product between the signal and the wavelets $\psi_{j,k}$ to be very high for a few $(j,k)$ combinations, and very low for the rest.

Daubechies wavelets are available in even-numbered kernel sizes ranging from two to 20 (Burrus et al., 1997; Chui, 1997). Larger kernels ensure a better frequency-resolution, but imply a worse time-resolution and produce problems in the construction of matrices $\mathbf{A}_m$ and $\mathbf{B}_m$, as some rows will get truncated very early in the procedure ($j$-levels). In our case, we used the eight-element kernel of Daubechies wavelets as a trade-off solution between frequency and time-resolution (Table 1).

DWT computation can be even faster than FFT (Strang, 1993). The calculation of the DWTs of 300 64-point data vectors by a non-compiled, non-optimized algorithm took less than 3 s in a 150 MHz Pentium with Windows98. Furthermore, our choice for the analyzing mother wavelet (Daubechies-8) was not optimized *in the sense of best fit* for extracellular neurophysiological spike profiles. By using an optimized mother wavelet and hardware implementation of the DWT, the WSC method could be applied on-line for real-time spike classification. On a historical note, it should be mentioned that a hardware implementation of the WSC method was anticipated 35 years ago by Glaser and Marks (cited in Schmidt, 1984b). They developed a device that used a three-channel transversal filter in a tapped delay line to provide a very crude time–frequency feature extraction of the spike profile.

Wavelet analysis has been introduced in biomedical signal analysis from the very beginning of the 'wavelet revolution', in the early 1990s (Aldroubi and Unser, 1996). Most applications in literature relate to image analysis (Aldroubi and Unser, 1996) and the electrophysiology of the cardiovascular (Akay et al., 1994; Dickhaus and Heinrich, 1996) and nervous systems, especially in evoked potential studies (Bertrand et al., 1994; Samar, 1999). Zouridakis and Tam (1997) presented a wavelet-based spike classification algorithm that concentrated on the problem of spike overlapping. Their method used a shift-independent, dyadic, semidiscrete wavelet transform in an adapted filter-bank al-

gorithm that omitted the downsampling features. In a recent paper (Zouridakis and Tam, 2000), the authors concentrated on the creation of accurate spike templates for template-based, fuzzy clustering. Fuzzy algorithms have the advantage of permanent adaptability, as they permit a fractional membership of spikes to many different classes, instead of a crisp membership value (1 = yes for one particular class, 0 = no for all others). Our method, as presented, does not use templates in the sense of representative spike profiles for each class or cluster with which the spikes to be sorted are compared. Still, fuzzy algorithms can be used for the automation of several tasks of the WSC method, for example the determination of classification potential for selecting separating $c_{j,k}$ coefficients. The clustering, which in the WSC method is done manually in a graphic interface, might also be automated by fuzzy clustering based on Euclidean differences between DWT vectors, as described by Zouridakis and Tam (2000), and require minimal to no user intervention. Such an automation might prove useful in the case of long recordings, with gradual changes in the spike profiles generated by every neuron. Up until now, the WSC method has already proven to perform fairly well in the presence of a $\pm 10\%$ amplitude variation (data not shown). Although DWT coefficients are reportedly sensible to small temporal jitters, the WSC method has been shown to outperform the RFS and PCA methods in the presence of a temporal jitter of up to $\pm 2$ data points ($= 0.1$ ms). Thus, the noise-resistance of wavelet-based classifying algorithms demonstrated by our work and the ideas of Zouridakis and Tam (1997, 2000) could be merged into a powerful spike sorter capable of both classifying extremely similar spike profiles and deconvolving superposition events into their constitutive elements.

Finally, due to its peculiar view of data in terms of a linear sum of features of different time scales and positions, wavelet analysis is particularly suited to unraveling the underlying physiology that generates neural signals. EEG and evoked potential modeling have been attempted with diverse mother wavelets and WT algorithms (Demiralp et al., 1999; Raz et al., 1999; Samar, 1999). Still, in extracellular recordings, specific time–frequency coefficients could likewise reveal certain functional features of a single generating cell, such as specific ionic channel subtypes with different inactivation constants, resulting in a spike classifying tool that exploits physiological rather than descriptive time-domain parameters. Further study is indicated to select the best algorithm for this application (like wavelet packets, with improved frequency-resolution at higher frequencies, or shift-invariant semidiscrete wavelet transforms) and to establish criteria for mother wavelet selection. More importantly, the WSC method might stimulate the rethinking of at least some parameters of physiology in terms of time–frequency coefficients.

## Acknowledgements

This research was partially supported by grant # 1990045 from Fondecyt. We acknowledge Dr Julio Alcayaga for the use of his electrophysiological data, Diane Greenstein for her editorial assistance, the support of the Laboratorio de Neurobiología y Biología del Conocer and the encouragement and interest of Dr Jorge Mpodozis. The computer code (IGOR) implementing the WSC procedure will be available upon request (letelier@uchile.cl).

## References

Akay M, Akay YM, Cheng P, Szeto HH. Time–frequency analysis of the electrocortical activity during maturation using wavelet transform. Biol Cybern 1994;71:169–76.

Aldroubi A, Unser M, editors. Wavelets in Medicine and Biology. Boca Raton, FL: CRC Press, 1996.

Bertrand O, Bohorquez J, Pernier J. Time–frequency digital filtering based on an invertible wavelet transform: an application to evoked potentials. IEEE Trans BME 1994;41:77–88.

Burrus CS, Gopinath RA, Guo H. Introduction to Wavelets and Wavelet Transforms: A Primer. Engelwood Cliffs, NJ: Prentice Hall, 1997.

Chui CK. Wavelets: A Mathematical Tool for Signal Analysis. SIAM Monographs on Mathematical Modeling and Computation. Philadelphia, PA: SIAM, 1997.

Demiralp T, Ademoglu A, Schürmann M, Basar-Eroglu C, Basar E. Detection of P300 waves in single trials by the wavelet transform. Brain Language 1999;66:108–28.

Dickhaus H, Heinrich H. Classifying biosignals with wavelet networks. A method for noninvasive diagnosis. IEEE-EMB 1996;Sept/Oct:103–11.

Eggermont J, Epping W, Aertsen A. Stimulus dependent neural correlations in the auditory midbrain of the grassfrog (*Rana temporaria* L.). Biol Cybern 1983;47:103–17.

Gray CM, Maldonado PE, Wilson M, McNaughton B. Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. J Neurosci Methods 1995;63:43–54.

Meyer Y. Wavelet analysis book report. Bull Am Math Soc (New Series) 1993;28:350–60.

Oghalai JS, Street WN, Rhode WS. A neural network based spike discriminator. J Neurosci Methods 1994;54:9–22.

Papoulis A. The Fourier Integral and Its Applications. New York: McGraw-Hill, 1962.

Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical Recipes in C: The Art of Scientific Computing. Cambridge: Cambridge University Press, 1993.

Raz J, Dickerson L, Turetsky B. A wavelet packet model of evoked potentials. Brain Language 1999;66:61–88.

Salganicoff M, Sarna M, Sax L, Gerstein GL. Unsupervised waveform classification for multi-neural recordings: a real-time, software based system. I. Algorithms and implementation. J Neurosci Methods 1988;25:181–7.

Samar VJ. Wavelet analysis of neuroelectric waveforms. Brain Language 1999;66:1–6.

Samar VJ, Boardikar A, Rao R, Swartz K. Wavelet analysis of neuroelectric waveforms: a conceptual tutorial. Brain Language 1999;66:7–60.

Sarna MF, Gochin P, Kaltenbach J, Salganicoff M, Gerstein GL. Unsupervised waveform classification for multi-neuron recordings: a real-time, software based system. II. Performance comparison to other sorters. J Neurosci Methods 1988;25:189–96.

Schmidt EM. Instruments for sorting neuroelectric data: a review. J Neurosci Methods 1984a;12:1–24.

Schmidt EM. Computer separations of multi-unit neuroelectric data: a review. J Neurosci Methods 1984b;12:95–111.

Sinkkonen J, Tiitinen H, Näätänen R. Gabor filters: an informative way for analysing event-related brain activity. J Neurosci Methods 1995;56:99–104.

Strang G. Wavelet transforms versus Fourier Transforms. Bull Am Math Soc (New Series) 1993;28:288–305.

Trejo LJ, Shensa MJ. Feature extraction of event-related potentials using wavelets: an application to human performance monitoring. Brain Language 1999;66:89–107.

Zouridakis G, Tam DC. Multi-unit spike discrimination using wavelet transforms. Comput Biol Med 1997;27:9–18.

Zouridakis G, Tam DC. Identification of reliable spike templates in multi-unit extracellular recordings using fuzzy clustering. Comput Methods Programs Biomed 2000;61:91–8.