Fast Continuous Wavelet Transform Based on B-Splines

Arrate Muñoz, Raphaël Ertlé and Michael Unser
Biomedical Imaging Group,
Swiss Federal Institute of Technology Lausanne
CH-1015 Lausanne EPFL, Switzerland

ABSTRACT

The Continuous Wavelet Transform (CWT) is an effective way to analyze nonstationary signals and to localize and characterize singularities. Fast algorithms have already been developed to compute the CWT at integer time points and dyadic or integer scales. We propose here a new method that is based on a B-spline expansion of both the signal and the analysis wavelet and that allows the CWT computation at arbitrary scales. Its complexity is $O(N)$, where N represents the size of the input signal; in other words, the cost is independent of the scale factor. Moreover, the algorithm lends itself well to a parallel implementation.

1. INTRODUCTION

The Continuous Wavelet Transform (CWT) of a signal f with the wavelet ψ is defined as

$$W_\psi f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x) \psi\left(\frac{b-x}{a}\right) dx = \frac{1}{\sqrt{a}} \left\langle f(\cdot), \psi\left(\frac{b-\cdot}{a}\right)\right\rangle. \tag{1}$$

It can be interpreted as the correlation of the input signal with a time-reversed version of ψ rescaled by a factor a. Typical applications of the CWT are the detection and characterization of singularities,[1] fractal analysis,[2] noise reduction[3] and the analysis of biomedical signals.[4]

Fast algorithms have already been developed to compute the CWT at integer time points and dyadic[2] or integer scales.[5] Despite their speed, these methods may not be fine enough for some applications because a dyadic scale progression is associated with an octave subband decomposition. Our purpose here is to develop a novel and fast B-spline-based algorithm for the computation of the CWT at any real scale a and integer time-localization b.

2. FAST CONTINUOUS WAVELET TRANSFORM

2.1. Description

Our method takes advantage of the convolution properties of B-splines. Both the mother wavelet and the input signal are represented in a spline basis. The wavelet at scale a is

$$\psi\left(\frac{x}{a}\right) = \sum_{k=-K}^{K} d_k \beta^{n_1}\left(\frac{x}{a} - k\right), \tag{2}$$

where β^{n_1} denotes the centered B-spline of degree n_1. Likewise, the continuous input signal $f(x)$ is represented by its spline interpolant which is in a one-to-one relation with the discrete input samples $f(k)$. Thus, we have

$$f(x) = \sum_{k\in\mathbb{Z}} c_k \beta^{n_2}(x-k), \tag{3}$$

where the interpolation coefficients c_k are calculated as shown elsewhere.[8]

Here, we only describe the algorithm without giving its derivation, since the latter is already available in.[9] Our method provides an exact evaluation of the convolution integral (1) when the wavelet and the signal are given by (2) and (3), respectively. The intuitive idea is that one can convolve a signal with a B-spline (of arbitrary size) by first

computing its (n_1+1)-fold integral, and then by taking the (n_1+1)th finite differences of the result. The fist step of the algorithm is to calculate the (n_1+1)-fold integral of the interpolation coefficients c_k

$$g = \Delta^{-(n_1+1)} * c, \tag{4}$$

where Δ^{-1} is the inverse finite-differences operator defined as $\Delta^{-1} = \sum_{n\geq 0} \delta(x-n)$.

The second step is to compute the inner products

$$W_\psi f(a,b) = \sum_{k=-K}^{K+n_1+1} \sum_{i=0}^{n_1+n_2+1} g_{i+i_0} w_{a,b}(k,i), \tag{5}$$

where

$$w_{a,b}(k,i) = p_k \beta^{n_1+n_2+1}(b - ak - i - i_0 + \tau) \tag{6}$$

is a filter mask (which we can store in a look-up table); $i_o = \lceil b - a(k+l) + \tau - \frac{n_1+n_2+2}{2} \rceil$ is the first meaningful index in the sum over i that is computed using the compact-support property of B-splines, $\tau = (a-1)\left(\frac{n_1+1}{2}\right)$ is a time shift and

$$p(k) = \frac{|a|}{a^{n_1+\frac{3}{2}}}(d * q)(k) = \frac{|a|}{a^{n_1+\frac{3}{2}}} \sum_{l=0}^{n_1+1} d(k-l)q(l)$$

where $q(k) = \binom{n_1+1}{k}(-1)^k$.

In practice, we are typically interested in the values of b that correspond to the time locations of the original samples, that is, for integer b. Then, we can use the fact that $w_{a,b} = w_{a,0}$ to reduce the dimension of the look-up table. The algorithm given in (5) is then equivalent to a discrete convolution. The filter w_a can be seen as a kind of modified 'à trous' filter. The CWT computation consists in filtering the coefficients g_i with $(2K+n_1+2)$ 'clusters' of length (n_1+n_2+2), each cluster being separated from its neighbors by a distance a.

2.2. Fast Implementation

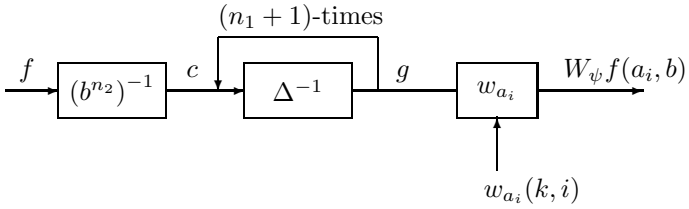


Figure 1. Schematic representation of the fast wavelet transform. $(b^{n_2})^{-1}$: Computation of the interpolation coefficients c. $\Delta^{-(n_1+1)}$: Calculation of the (n_1+1)-fold integral of c. $w_{a_i}(k,i)$: Look-up table calculation where $k \in [-K, K+n_1+1]$, $i \in [0, n_1+n_2+1]$ and $a_i \in [a_1, a_N]$. N is the number of scales. w_{a_i}: Filtering with the mask calculated for each scale. $W_\psi f(a_i, b)$: Wavelet transform of f for scale a_i at position b.

Let us describe now the fast algorithm based on the expansion (5). In the initialization step, the B-spline expansion coefficients c_k of the sampled signal $f(x)$ are calculated, and the running-sum operator Δ^{-1} is applied (n_1+1)-times for implementation details see.[10] The intermediate result g_i does not depend on the scale a. For a given scale a, we compute the weights w_a and store them in the 2-D matrix of dimensions $(2K+n_1+2)\times(n_1+n_2+2)$ (look-up table). These values are then convolved with the precomputed sequence g_i. The values w_a and the inter-cluster distance for the filtering depend on a, but the computational complexity is constant and does not depend on a. Note that the independence between scales allows for a straightforward parallel implementation (see Figure 1).

3. RESULTS AND DISCUSSION

Here, we discuss the implementation of our fast CWT algorithm and compare its execution time with a FFF-based implementation. As example of application, we show the analysis of biomedical signals.

3.1. Comparison with FFT-Based Computation

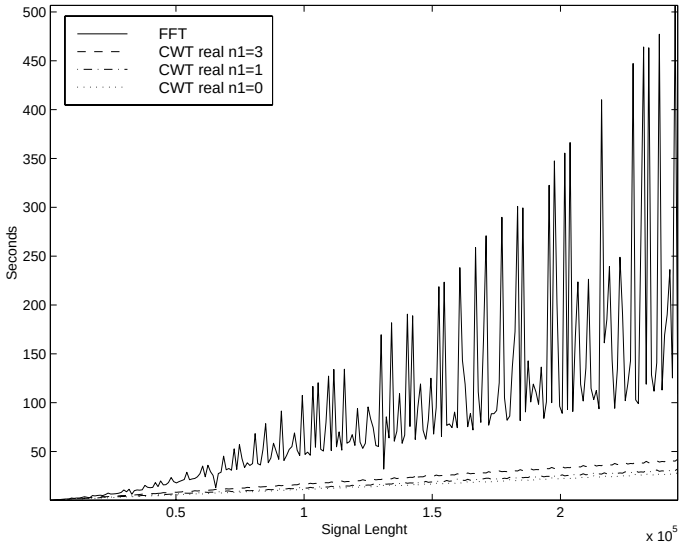


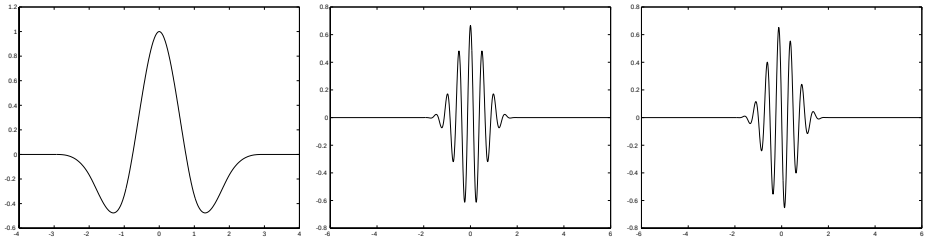
Figure 2. Comparison in computational time between the FFT and our B-spline-based method to calculate the CWT.

As mentioned in the introduction, the FFT has an overall $O(N \log N)$ complexity and is therefore asymptotically slower than our method which has an $O(N)$ complexity. This can be observed from the experimental comparison of the computation time shown in Figure 2. We see that, for long input signals, our method is indeed faster. The interpolation degree for the input signal was zero. The CWT was computed over four octaves with 12 scales per octave. The wavelet was the second derivative of the quintic, quartic and cubic spline, respectively (see Figure 3). The FFT-based method used a radix-2 algorithm when the signal length was a power of 2 and a mixed-radix method for other signal lengths (MATLAB's FFT algorithm). The time required to compute the wavelet in the time domain before its FFT computation was neglected. A parallel implementation for the scale-dependent part of each algorithm would speed up the computations.

3.2. Example of Application

We have applied our method to the analysis of bowel movements. A magnetically active capsule was swallowed and its gastrointestinal transit was monitored. The measures consisted in its three spatial coordinates and the angles that describe its orientation.[11]

Figure 4 (a) corresponds to the x-coordinate of the stomach signal. The sampling time was 70 ms. We have analyzed it using both real (Figure 4 (b)) and complex CWT (Figure 4 (c)) for cubic spline interpolation of the input signal. In this application, the wavelet used was the second derivative of a quintic spline (Figure 3 (a)). There, we observe mainly the band around the scales $36 - 40$ corresponding to the breathing of the patient. We choose to calculate the complex CWT an extension of the method proposed for Unser *et al.*[12] for integer scales. The analysis wavelet is the Gabor-like wavelet $\beta^3(x)e^{-4\pi xj}$ (Figure 3 (b) and (c)). Using complex analysis (Figure 4 (c)), we have discovered three relevant frequency bands: First, the breathing with a period close to 3 s; then, two more bands with periods of 12 and 20 s due to the contractions of the stomach.



(a) (b) (c)

Figure 3. (a) Second derivative of the quintic spline wavelet. Gabor-like wavelet: $\beta^3(x)e^{-4\pi xj}$ (b) real component; (c) imaginary component.

4. CONCLUSION

We have presented a novel B-spline-based CWT algorithm that is able to compute the CWT at any real scale, making it possible to use an arbitrary scale progression. Its complexity is $O(N)$, where N represents the size of the input signal, the same complexity as with the most efficient wavelet algorithms for dyadic or integer scales. The overall operation count only depend on the wavelet shape and on the degrees of the B-splines basis on which the wavelet and input signal are described, but is independent of the value of the scale. Moreover, the algorithm lends itself well to a parallel implementation as it is not iterative across scales. The price to pay for the generality of this algorithm is that the leading constant of the $O(N)$ complexity can be large. Thus, it only really starts paying off when the size of the signal is large; say ($N \geq 1000$ samples). For smaller sizes, it may be preferable to use a simpler FFT-based approach.

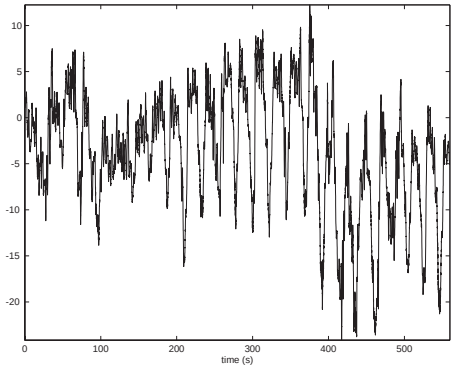
5. ACKNOWLEDGEMENTS

This work was funded by the Swiss National Science Foundation, grant #2100-053540. Moreover, the authors would like to thank R. S. Popovic and V. Schlageter for giving us the permission to use the physiological signal.

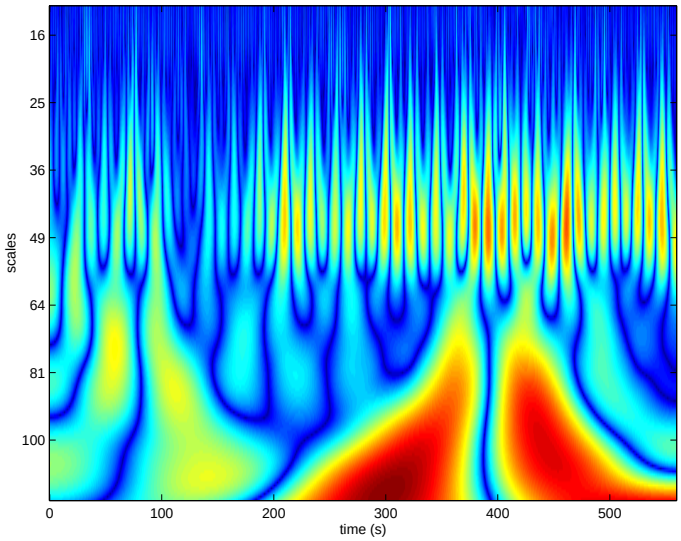
REFERENCES

1. A. Munteanu, J. Cornelis, P. De Muynck, A. Bezerianos, and P. Cristea, "Accurate detection of coronary arteries with the continuous wavelet transform," *Computers in Cardiology*, vol. 24, pp. 601–604, 1997.
2. S. Mallat, *A wavelet tour of signal processing*, Academic Press, San Diego, USA, 1998.
3. J. M. Lewis and C. S. Burrus, "Approximate continuous wavelet transform with an application to noise reduction," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998, vol. 3, pp. 1533–1536.
4. H. Yoshida, B. Keserci, D. D. Casalino, O. Ozturk A. Coskun, and A. Savranlar, "Segmentation of liver tumors in ultrasound images based on scale-space analysis of the continuous wavelet transform," in *IEEE Ultrasonics Symposium*, 1998, pp. 1713–1716.
5. O. Rioul, "Fast algorithms for the continuous wavelet transform," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 1991, vol. 3, pp. 2213–2216.
6. M. J. Vrhel, C. L. Lee, and M. Unser, "Rapid computation of the continuous wavelet transform by oblique projections," *IEEE Transactions on Signal Processing*, vol. 45, no. 4, pp. 891–900, April 1997.
7. M. Unser, A. Aldroubi, and S. J. Schiff, "Fast implementation of the continuous wavelet transform with integer scales," *IEEE Transactions on Signal Processing*, vol. 42, no. 12, pp. 3519–3523, December 1994.
8. M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Transactions on Signal Processing*, vol. 16, pp. 22–38, November 1999.
9. A. Muñoz, R. Ertlé, and M. Unser, "Continuous wavelet transform with arbitrary scales and $O(N)$ complexity," *to be submitted to Signal Processing*.
10. A. Muñoz, T. Blu, and M. Unser, "Least-squares image resizing using finite differences," *in press for IEEE Transactions on Image Processing*.

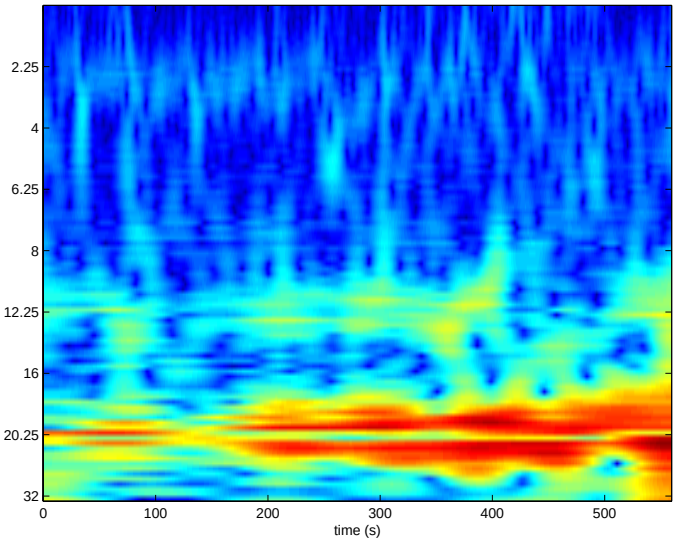
11. V. Schlageter, P. A. Besse, R. S. Popovic, and P. Kucera, "Tracking system with five degrees of freedom using a 2-D array of Hall sensors and a permanent magnet," *Sensors and Actuators*, vol. 2951, pp. 1–6, 2001.
12. M. Unser, "Fast Gabor-like windowed Fourier and continuous wavelet transforms," *IEEE Signal Processing Letters*, vol. 1, no. 5, pp. 76–79, May 1994.



(a)



(b)



(c)

Figure 4. (a) Signal of the displacement of a magnet within the digestive track. (b) Real CWT using the wavelet in Figure 3-(a). (c) Complex CWT using the Gabor-like wavelet shown in Figures 3 (b)-(c).