Richard Kronland-Martinet
LMA C.N.R.S and Faculté des Sciences de Marseille-Luminy
31 chemin J. Aiguier
13402 Marseille Cedex 9
France

The Wavelet Transform for Analysis, Synthesis, and Processing of Speech and Music Sounds

Introduction

The wavelet transform is a recent method of signal analysis and synthesis (Grossmann and Morlet 1984; Grossmann et al. 1987). It analyzes signals in terms of *wavelets*—functions limited both in the time and the frequency domain. In comparison, the classical Fourier analysis method analyzes signals in terms of sine and cosine wave components that are not limited in time.

The wavelet transform is related to granular analysis/synthesis, first suggested by Gabor (1946). Granular synthesis has been implemented by Roads (1978) and Truax (1988). Rodet (1985) and Liénard (1984) use adapted grains for speech signals; however, these implementations do not attempt to reconstruct an arbitrary given signal.

The Gabor method uses an expansion of a function into a two-parameter family of *elementary wavelets* that are obtained from one *basic wavelet* by shifts in the time variable and in the frequency variable (Fig. 1a). The practical limitation of this procedure can be seen in the case where one of the signals to be analyzed is a short, high-frequency transient. In this case, in the process of reconstruction, it is necessary to sum over many terms of rapidly varying phase. This leads to instability in the numerical computations.

The wavelet transform uses a related but different procedure. It decomposes an arbitrary function again into a two-parameter family of elementary wavelets that are obtained by shifts in the time variable but also by dilations (or compressions) that act both on the time and the frequency variables (Fig. 1b). So, contrary to the Gabor method, the number of cycles does not change. Thus the new method is easier to implement; it permits a better convergence of the reconstruction formulas.

Editor's note: A soundsheet with audio examples of the wavelet transform will appear in the next issue, Volume 13, Number 1, Spring 1989.

Computer Music Journal, Vol. 12, No. 4, Winter 1988,

What Is an Analyzing Wavelet?

We have great latitude in the choice of the analyzing wavelet used for analysis. However, it cannot be completely arbitrary and the conditions to be fulfilled are mathematically well-defined (Grossmann and Morlet 1984).

Let $g(t)$ be an analyzing wavelet. The requirements are as follows:

(i) $g(t)$ is absolutely integratable and square integratable (the latter condition means it has a finite energy):

$$\int |g(t)|\, dt < \infty \quad \text{and} \quad \int |g(t)|^2\, dt < \infty. \tag{1}$$

(ii) If $\hat{g}(\omega)$ represents the Fourier transform of $g(t)$, then

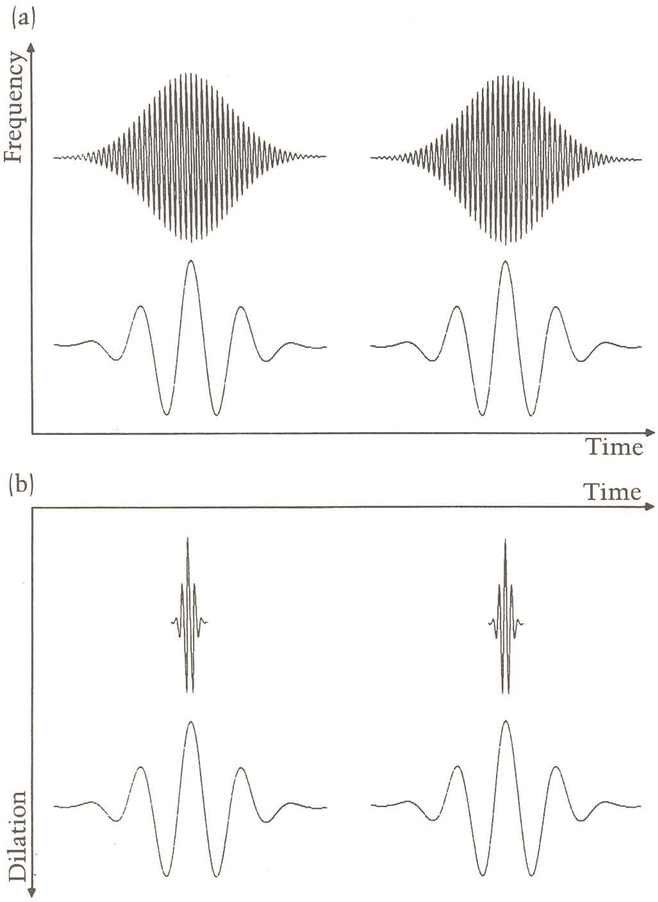
$$\int |\hat{g}(\omega)|^2/\omega \; d\omega < \infty. \tag{2}$$

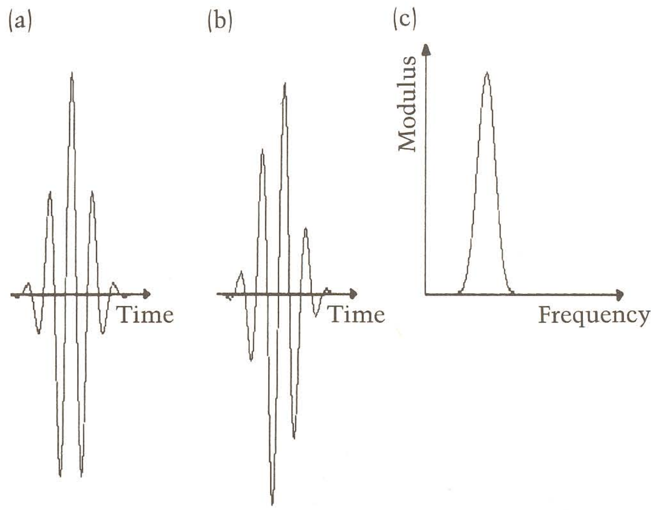
In practice, this requires that $g(t)$ have zero mean value (no DC bias): $\hat{g}(0) = 0$ or $\int g(t)\, dt = 0$.

In the case of speech and music-sound analysis, it is convenient to extract from the analysis some information about the energy distribution and phase behavior in the wavelet transform representation. This can easily be done by using a complex-

Fig. 1. Elementary wavelets used for the Gabor expansion in the time-frequency domain (a) and for the wavelet transform in the time-dilation domain (b). The number of cycles does not change.

Fig. 2. An example of analyzing wavelet: real part (a); imaginary part (b); Fourier transform (c). Notice the localization both in the time and frequency domains.





valued wavelet. So, we make here an additional assumption:

(iii) $g(t)$ contains only positive frequency components. $\hat{g}(\omega) = 0$ for $\omega < 0$. So the real and imaginary parts of $g(t)$ are the Hilbert transform of each other.

These are the minima requirements that $g(t)$ has to satisfy in order to be an analyzing wavelet. In practice, one often also requires a concentration of $g(t)$ and $\hat{g}(\omega)$ not too far from the limit imposed by the uncertainty principle. As an example, the wavelet used for the computation of the numerical examples below is Fig. 2

$$\hat{g}(\omega) = K \cdot \exp(-(\omega - \omega_0)^2/2) + \text{small corrections.} \quad (3)$$

The small corrections are numerically negligible but ensure that $g(t)$ satisfies the formal conditions for an analyzing wavelet. The corresponding time expression of the analyzing wavelet is

$$g(t) \approx C \cdot \exp(-t^2/2) \cdot \exp(i\ \omega_0\ t). \quad (4)$$

The Wavelet Transform

Now, consider an analyzing wavelet. We want to expand an arbitrary signal into contributions that all have the same shape as our wavelet. We obtain these contributions by shifts and dilations of the original wavelet.

Let $g(t)$ be the analyzing wavelet, a the dilation parameter $(a > 0)$ and b the shift parameter. We define a two-parameter family of wavelets $g_{a,b}(t)$, which can be mapped on a *shift-dilation plane* (Fig. 1b):

$$g_{a,b}(t) = g((t - b)/a).$$

The wavelet transform of an arbitrary signal $s(t)$ is now defined on every point (b, a) in the shift-dilation plane. Just take the associated wavelet $g_{a,b}(t)$, then do the scalar product of $g_{a,b}(t)$ with the signal. The wavelet transform is a complex-valued function in our case, since the wavelet is complex-valued.

So the wavelet transform of the signal $s(t)$, with respect to the wavelet $g(t)$, is the function $S(b, a)$ on the shift-dilation plane defined by

Fig. 3. Sampling grid in the time-dilation domain. This grid allows an arbitrarily precise reconstruction of the signal.

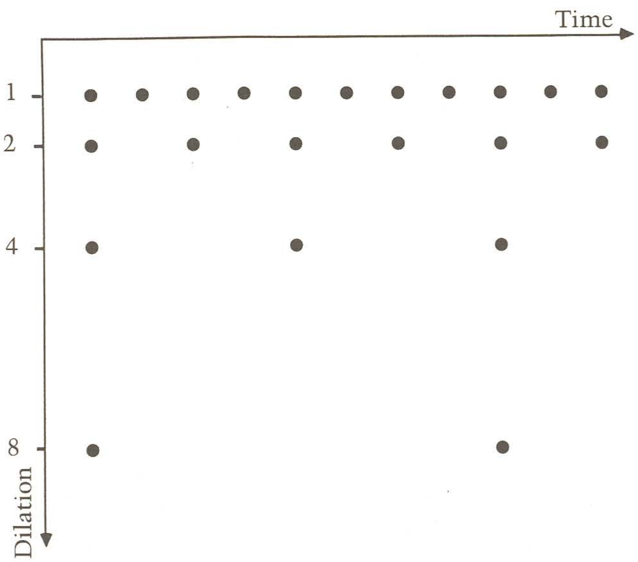
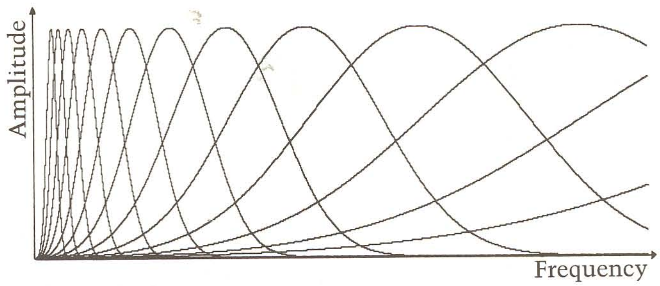


Fig. 4. Response of a bank of filters corresponding to dilated wavelets. The ration δf/f does not change.



$$S(b, a) = (1/\sqrt{a}) \int \bar{g}((t - b)/a)\, s(t)\, dt$$
(the bar denotes the complex conjugate). (5)

An important point is the ability to reconstruct the signal $s(t)$ from its transform $S(b, a)$. Actually, there is no loss of information when we go from the wavelet transform to the resynthesized signal. An exact reconstruction formula is given by

$$s(t) = (1/c_g) \int\int (1/\sqrt{a})\, g((t - b)/a)\, S(b, a)\, (1/a^2)\, da\, db \quad (6)$$

(c_g is a constant depending only on the wavelet g chosen).

There exist many other formulas for resynthesis. An important one, as we shall see in the next section, only needs a simple integration (summation) of the coefficients:

$$s(t) = K_g \int S(t, a)\, (1/a^{3/2})\, da. \quad (7)$$

These formulas involve the values of the wavelet transform on a continuous plane (b, a). The mathematics also give important practical results concerning the possibility of discretization, replacing the integration by a discrete summation and the storage of the transform. Other reconstruction formulas exist that involve only the values of $S(b, a)$ on a suitable discrete grid (Fig. 3).

The Wavelet Transform and Hearing

If one uses a wavelet similar to that shown in Fig. 2a, then the analysis process is somewhat similar to the analysis performed by hearing. This analogy derives from the fact that the wavelet transform performs an analysis with $\delta f/f$ constant, which is related to the analysis done by the auditory system. Of course, the condition is that the analyzing wavelet used looks like the response at a particular point on the basilar membrane to a short impulse, or more simply that $\delta f/f \approx 1/3$. (The critical band of the ear is about 1/3 octave in the middle range.)

Figure 4 shows the responses of a bank of filters corresponding to the wavelet family $g_{a,b}(t)$ with $a = 2^j$ (one filter per octave).

The frequency selectivity of these filters is better for low frequencies; their time selectivity is better for high frequencies. These characteristics allow us, for instance, to see beats on frequency components when they occur in the audio range. (Figs. 5 and 6).

Implementation of the Wavelet Transform

The method just described has been implemented with the real-time signal processor SYTER, designed by J. F. Allouis at INA-GRM and built by Digilog (Allouis and Mailliard 1981). This system consists of a host processor (DEC PDP-11/73) and a real-time signal processor (Fig. 7) that can calculate the wavelet transform by using a digital transverse filter (Fig. 8). The system allows also the reconstruction of signals from their transform with the possibility of performing modifications by altering the parameters after analysis and before resynthesis.

Fig. 5. Modulus of the wavelet transform of a two-frequency signal. Since the two components are far from each other, no interference occurs.

Fig. 6. The same as Fig. 5 with an inharmonic frequency ratio. One sees the interferences corresponding to beats for hearing.

Fig. 7. Schematic description of the SYTER processor.

Fig. 5

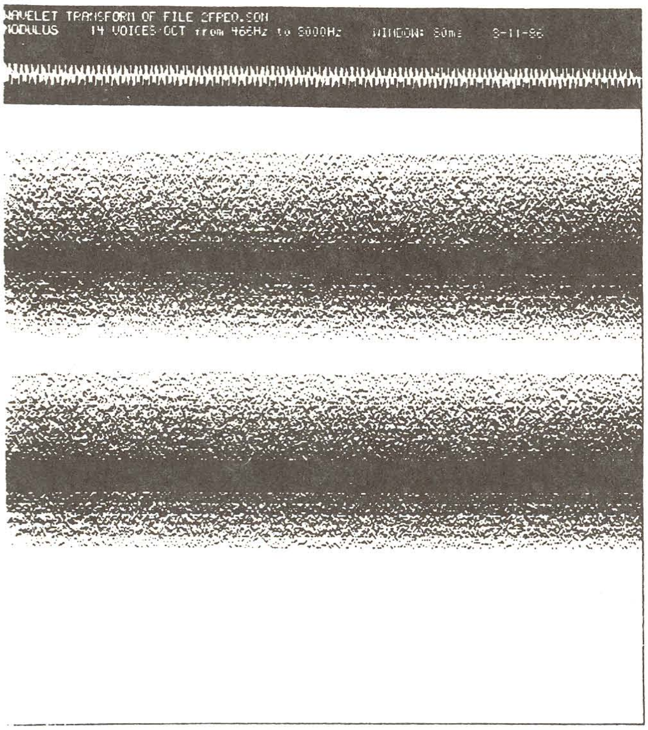


Fig. 6

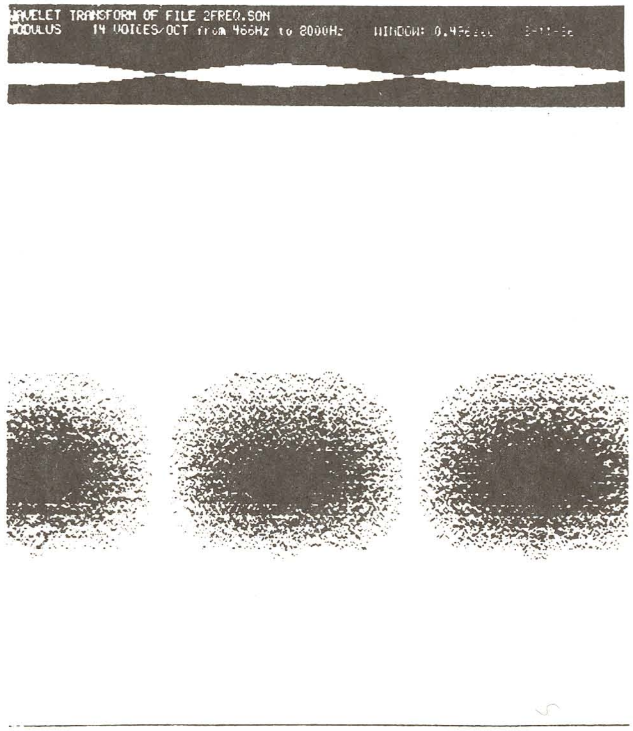


Fig. 7

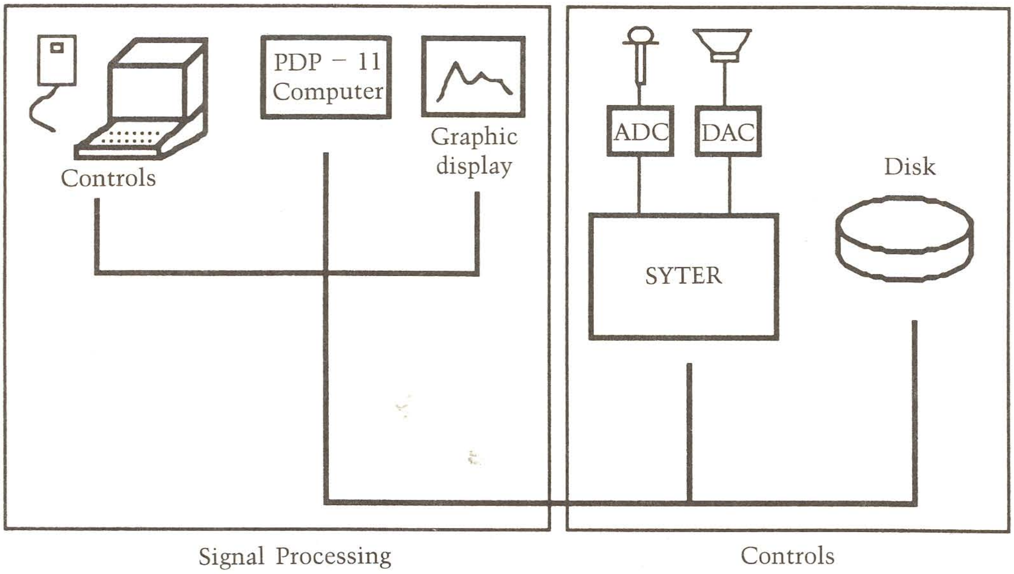


Fig. 8. *Structure of the transverse filter used for wavelet transformation.*

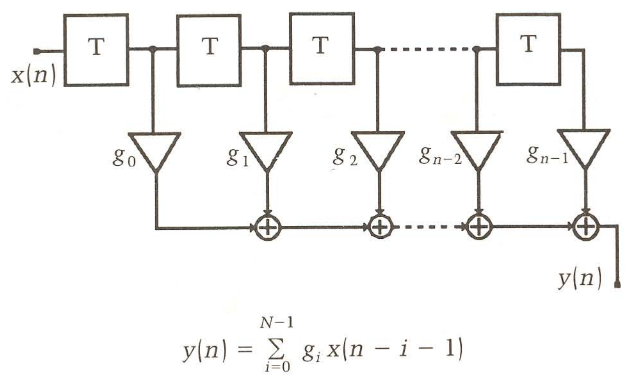
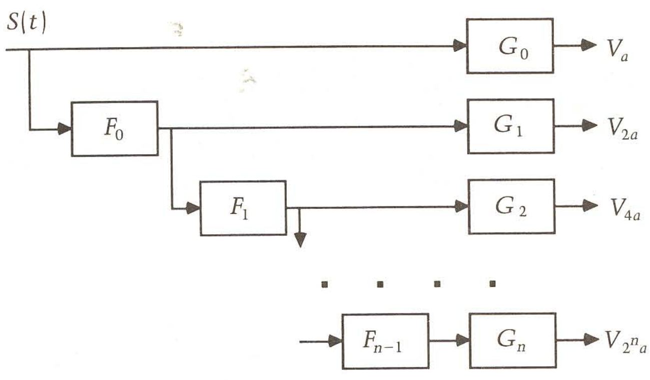


Fig. 9. *Schematic description of a real-time algorithm used for wavelet transformation.*



The discrete convolution described in Eq. 5 is

$$X(n) = \Sigma\, g_i\, S(n - 1 - i), \tag{8}$$

where $X(n)$ is the nth output sample, g_i the ith value of the wavelet, and $S(n)$ the nth sample of the input signal.

For any value of the dilation parameter a (for each "voice"), the host processor calculates the sample values $g_a(n)$ of the analyzing wavelet, and transmits them to the real-time signal processor, which then performs the transform in real time for each voice.

This algorithm can be hard to implement in real time if the analyzing wavelet is defined over a large number of sampled values (due to dilation). For some classes of wavelets, it is possible to compute the convolution with a fixed number of points for each octave by using a preliminary transverse filter acting on the signal (Holschneider et al. 1988). In general, this filter needs three or four nonzero coefficients. A diagram of this algorithm is given in Fig. 9. Here G_0 and F_0 represent respectively the discrete wavelet and the initial discrete impulse response of the filter. G_n and F_n are defined recursively from G_{n-1} and F_{n-1} by added zeros between each point.

A Look at Some Results

In this section, we present the first applications of a method to visualize characteristic features (the "acoustic signature") and to resynthesize and process speech and musical sounds, using a wavelet similar to that of Fig. 2.

Sound Analysis

The wavelet transform is localized both in the time and in the frequency domain. This allows us to analyze natural sounds with great precision (Kronland-Martinet, Morlet, Grossmann 1987). As an example, Figs. 10, 11, 12, and 13 are representations of complex-valued functions $S(b, a)$ on the open (b, a) half-plane. The positive b-axis points to the right and the Log(a)-axis points downward. It is useful to separately represent the modulus and the phase of the function $S(b, a)$. Both quantities are here represented by shades of gray obtained through appropriate random printing of black dots. The modulus display is quite similar to a sonogram, showing the frequency content as a function of time.

In some applications, it is possible to adapt the analyzing wavelet in order to extract the specific information. As an example, consider the musical staff of Fig. 14, and suppose we try to detect all the octave intervals by analyzing the acoustic signal. For this purpose, it is convenient to choose an analyzing wavelet related to the octave interval. We can use here wavelets that contain, in the frequency domain, two bumps, spaced an octave apart. Figure 15 represents the Fourier transform and the real and imaginary part of one of these wavelets. In this case, the maximum energy—corresponding to the larger correlation between the wavelet and the sig-

Fig. 10. Modulus of the transform of the first 16 msec of a clarinet sound. The frequency range is 50–8000 Hz.

Fig. 11. Phase picture corresponding to Fig. 10.

Fig. 12. Modulus of the transform of 16 msec of a sound signal taken from an old record. Notice the characteristic feature corresponding to the scratch.

Fig. 13. Phase picture corresponding to Fig. 12.

Fig. 10

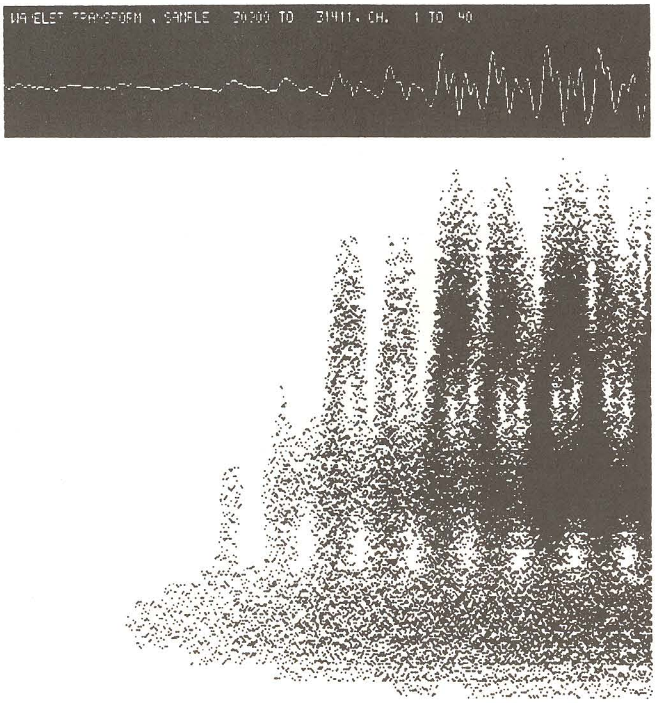


Fig. 11

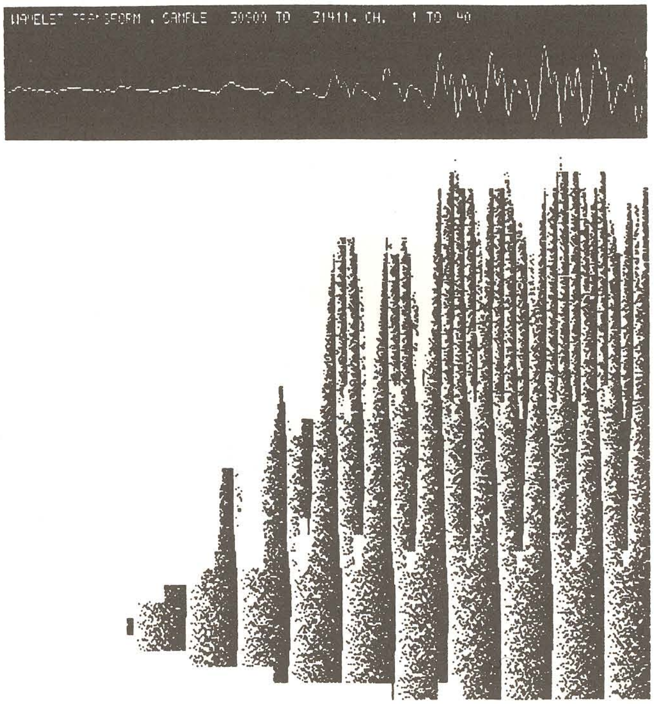


Fig. 12

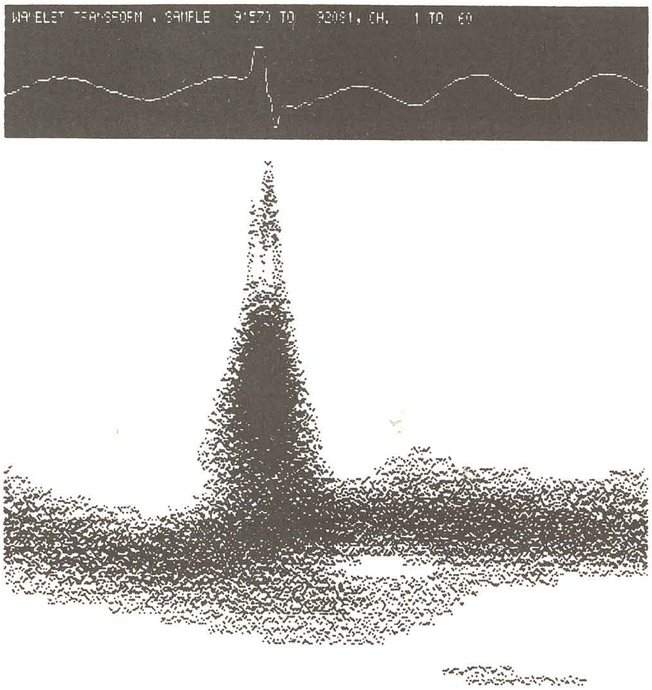


Fig. 13

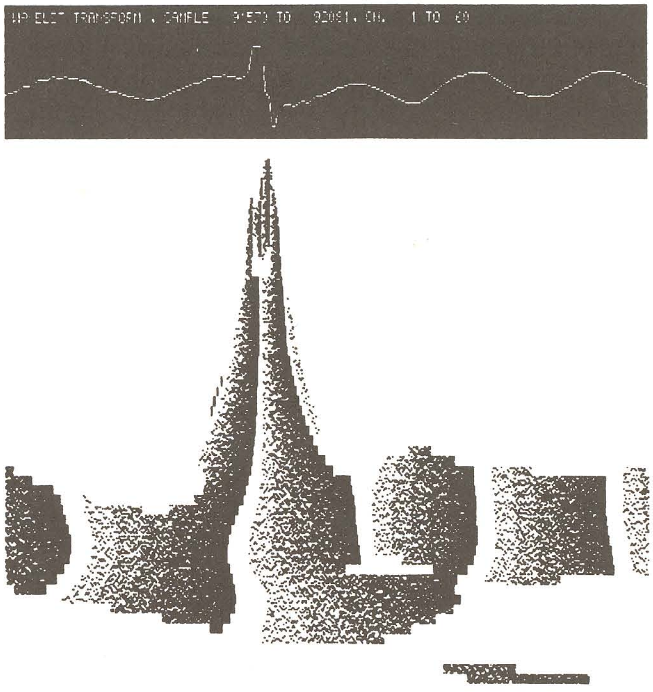


Fig. 14. An example of analyzing wavelet adapted to octave detection: real part (a); imaginary part (b); Fourier transform (c).

Fig. 15. Modulus of the wavelet transform of signal corresponding to the staff below. The wavelet used is given in Fig. 14. Maxima occur when octaves are played.

Fig. 16. Schematic representation of synthesis by summation of wavelets, corresponding to granular synthesis.

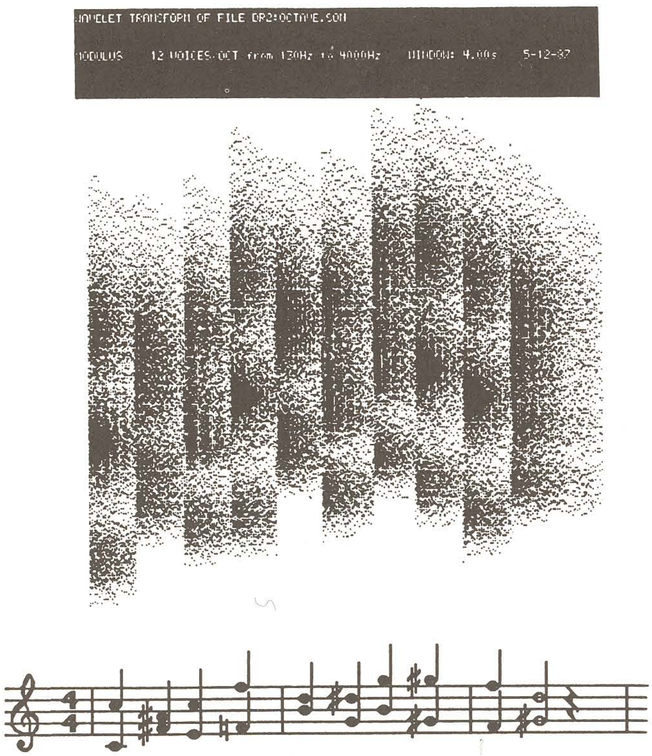
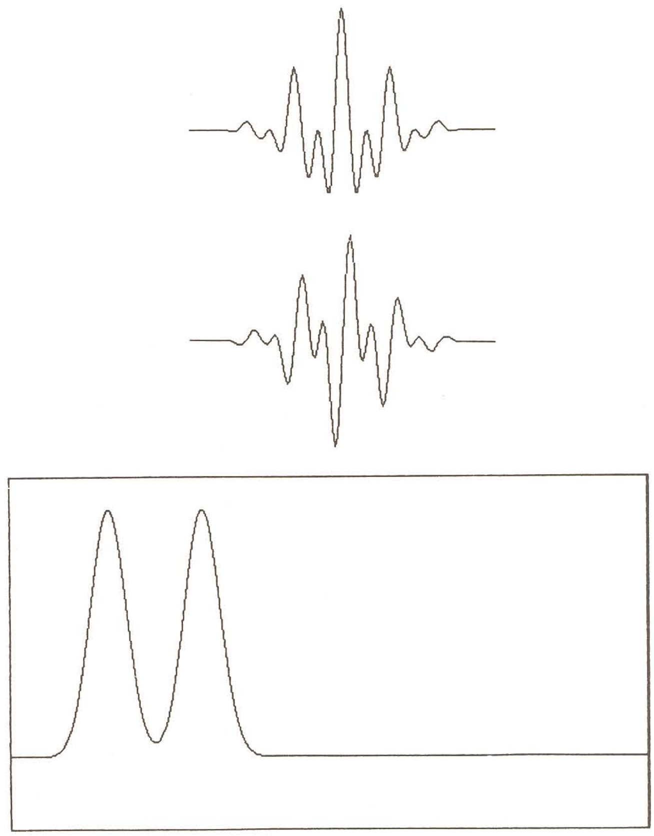


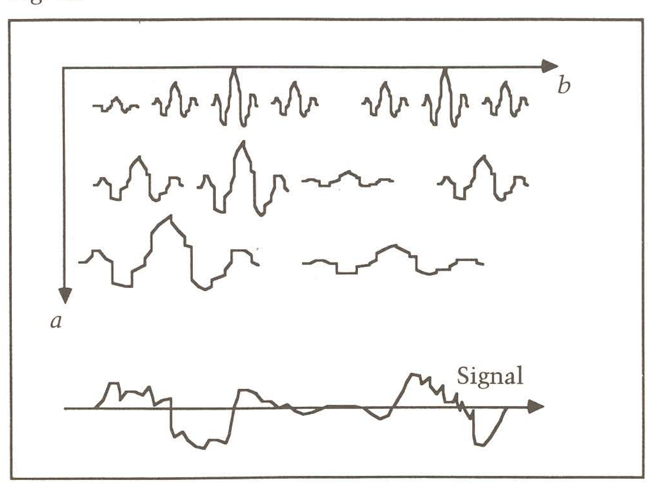
Fig. 15

$$g(t) \approx K.\ \exp(-t^2/2)\ [\exp(i\omega_0 t) + \exp(2i\omega_0 t)]$$



nal—appears as an increase of darkness when octave intervals are played (Fig. 14). Notice that in this case, the wavelet transform representation is no longer a true time-frequency representation (sonogram); we analyze the signal in terms of contributions that reveal octaves; it is really a time-scale representation and not a time-frequency one.

Fig. 16



Synthesis of Sounds

The SYTER processor lets us listen to the reconstructed signals. This enables us to check the acoustic relevance of the method. As we have seen, there are numerous formulas for signal reconstruction. Equation 6 suggests a granular synthesis technique. It consists of a summation of all the grains constituted by the dilated and shifted wavelets with a complex gain equal to the coefficients $S(b, a)$ obtained in the analysis (Fig. 16).

Equation 17 is easier to implement. It involves only the values of the wavelet transform at a given

Fig. 17. Elementary cell in a quasi-Music V notation (Mathews et al. 1969) used for the resynthesis. Phase and modulus are given by the coefficients of the wavelet transform.

Fig. 18. The instrument used for resynthesis. Each cell is given by Fig. 17.

Fig. 19. The phase picture corresponding to the wavelet transform of a single sine.

Fig. 20. Elementary cell in Music V notation used for resynthesis with frequency transposition effect (ratio = n).

Fig. 17

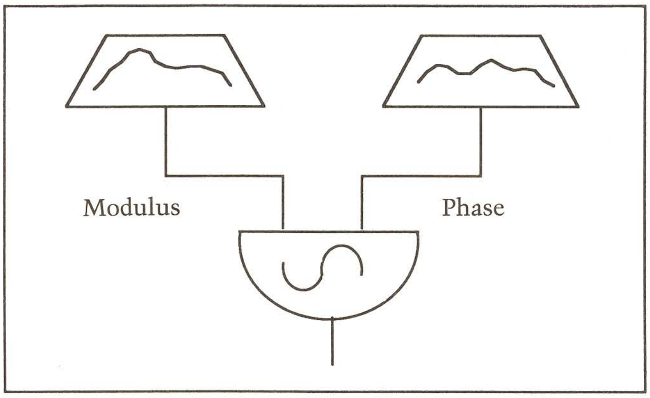


Fig. 18

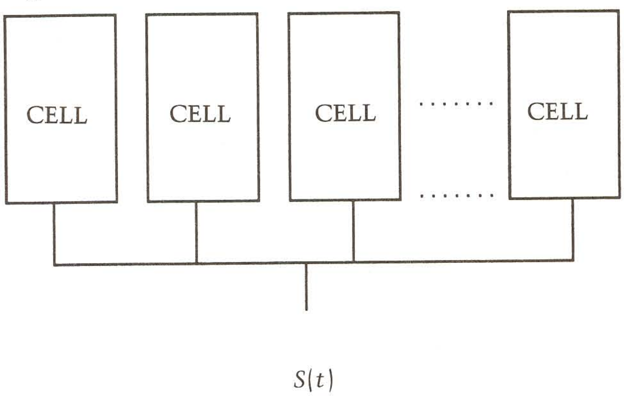


Fig. 19

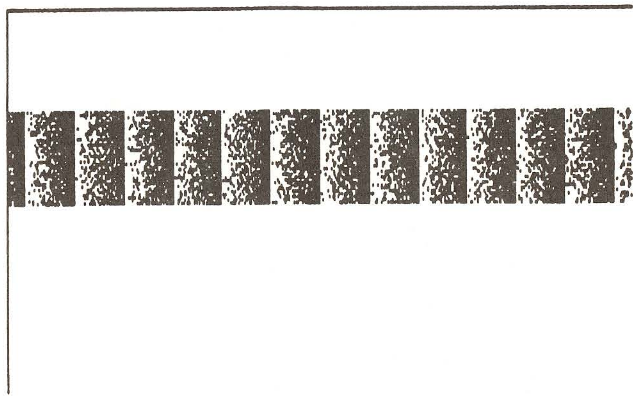
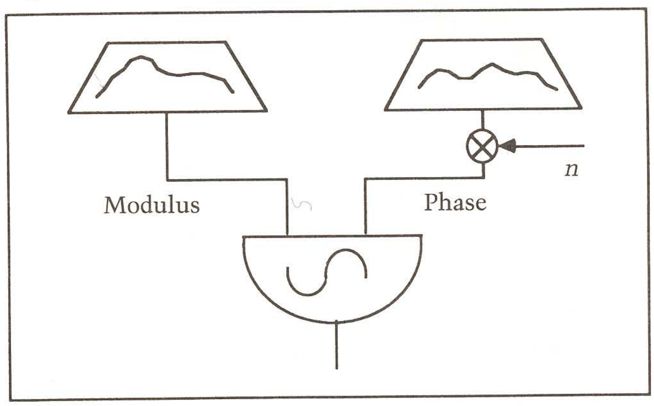


Fig. 20



time in order to reconstruct the corresponding value of the signal.

The discretization of Eq. 7 on a grid defined for scale parameters $a = 2^j$ (corresponding to an analysis at one voice per octave) can be used for the analyzing wavelet described next, and gives

$$s(t) = k_g \sum_{a=2^j} S(t, a) \cdot a^{-1/2}. \quad (9)$$

Since $S(t, a) = A_a(t) \cdot e^{i\varphi_a(t)}$, where $A_a(t)$ and $\varphi_a(t)$ represent respectively the modulus and the phase of the wavelet transform coefficients for a fixed-scale parameter a, the reconstructed signal can be obtained by

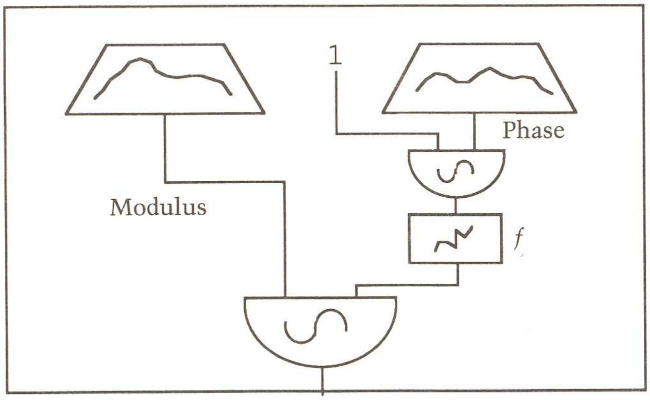
$$s(t) = k_g \sum_{a=2^j} A_a(t) \cdot \cos(\varphi_a(t)) \cdot a^{-1/2}. \quad (10)$$

This formula is in fact an additive synthesis model of the signal. For each voice, the amplitude modulation is given by the time function $A_a(t)$ and the phase modulation by $\varphi_a(t)$. The elementary cell is represented in Fig. 17, and the final instrument, obtained by some cells in parallel (typically 10), is shown in Fig. 18.

Modification of Real Sounds

The wavelet transform can modify signals in intimate ways that can be of interest in particular for computer music (Risset and Wessel 1982). Numerous modifications can be imagined, for example as with resynthesis using the phase vocoder (Moorer 1978; Dolson 1986).

Fig. 21. Elementary cell in Music V notation used for the resynthesis with frequency transposition effect with an integer ratio. (f is a Chebyshev polynomial.) This algorithm can perform multiple transpositions (for a brightness effect) when f is a sum of Chebyshev polynomials.



For example, let us look at the effect of transposing the signal in frequency without acting on its duration. This problem is difficult and the solution often depends on the signal itself. Let us take first the case of a simple signal, $x(t) = A \cdot \cos(\omega_0 t)$. Then

$$S(b, a) = K_a \int x(t)\, g(t - b)/a)\, dt = K_a\, \hat{g}(a\, \omega_0)\, \exp(ib\omega_0). \quad (11)$$

Here, the phase coefficient of the wavelet transform is given by $b\omega_0$ (Fig. 19).

We can transpose the frequency of the original signal from f to $n \cdot f$ by multiplying the phase $b\omega_0$ by n. Let $x_n(t)$ be the transposed and resynthesized signal. From Eq. 10 we obtain

$$x_n(t) = k_g \sum_{a=2^j} A_a(t) \cdot \cos(n\omega_0 t) \cdot a^{-1/2}. \quad (12)$$

We can approach the transposition effect for any arbitrary signal $s(t)$ by

$$s_n(t) = k_g \sum_{a=2^j} A_a(t) \cdot \cos(n\, \varphi_a(t)) \cdot a^{-1/2}. \quad (13)$$

The elementary cell corresponding to this algorithm is given in Fig. 20.

Notice that for n integers, we can use a waveshaping module (Arfib 1979) in order to construct $s_n(t)$. Indeed, let $T_k(x)$ be the kth order Chebyshev polynomial that verifies

$$T_k(\cos \varphi(t)) = \cos(k\, \varphi(t)). \quad (14)$$

We can thus multiply by k the phase coefficients, by using T_k as a waveshaping function with a fixed index equal to 1.

We can generalize this remark and perform a multiple transposition by using as waveshaping function a sum of Chebyshev polynomials,

$$f(x) = \sum C_k\, T_k(x), \quad (15)$$

and then construct

$$S(t) = k_g \sum_{a=2^j} A_a(t) \cdot f(\cos(\varphi_a(t)) \cdot a^{-1/2}. \quad (16)$$

If in Eq. 15 the coefficient $c_1 = 1$, we obtain a brightness effect of the original sound.

A schematic diagram of this algorithm is given in Fig. 21.

We can thus perform a number of sound modifications by altering the wavelet transform coefficients. Such possibilities include slowing down or speeding up the sound without pitch transpositions, time-varying filtering, and cross-synthesis between two sounds by resynthesis with the modulus information of one sound and the phase information of another sound.

References

Allouis, J. F., and B. Mailliard. 1981. "Simulation par ordinateur du studio électroacoustique et applications à la composition musicale." *Conférence des Journées d'Études, Festival International du Son Haute Fidélité.* Editions SDSA, pp. 43–55.

Arfib, D. 1979. "Digital Synthesis of Complex Spectra by means of Multiplication of Non-linear Distorted Sine Waves." *Journal of the Audio Engineering Society* 27(10): 757–768.

Dolson, M. 1986. "The Phase Vocoder: A Tutorial." *Computer Music Journal* 10(4): 14–27.

Gabor, D. 1946. "Theory of Communication." *Journal of the Institute of Electrical Engineers* 93: 429–441.

Grossmann, A., M. Holschneider, R. Kronland-Martinet, and J. Morlet. 1987. "Detection of Abrupt Changes in Sound Signals with the Help of Wavelet Transforms." *Inverse Problems: Advances in Electronics and Electron Physics.* Suppl. 19. Orlando, Florida: Academic Press, pp. 298–306.

Grossmann, A., and J. Morlet. 1984. *SIAM Journal of Mathematical Analysis* 15: 723–736.

Holschneider, M., R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. 1988. "A Real-time Based Algorithm for Signal Analysis with the Help of the Wavelet Trans-

form." Note CPT. Marseille, France: Université de Marseille-Luminy.
Kronland-Martinet, R., J. Morlet, and A. Grossmann. 1987. "Analysis of Sound Patterns through Wavelet Transforms." *International Journal of Pattern Recognition and Artificial Intelligence* (Special issue on expert systems and pattern analysis) (2): 97–126.
Liénard, J. S. 1984. "Speech Analysis and Reconstruction Using Short-Time, Elementary Waveforms." Note LIMSI-CNRS. Orsay, France: LIMSI-CNRS.
Mathews, M. V., et al. 1969. *The Technology of Computer Music.* Cambridge, Massachusetts: MIT Press.
Moorer, J. A. 1978. "The Use of the Phase Vocoder in Computer Music Applications." *Journal of the Audio Engineering Society* 26: 42–45.
Risset, J. C., and D. L. Wessel. 1982. "Exploration of Timbre by Analysis and Synthesis." In D. Deutsch, ed. *The Psychology of Music.* Orlando, Florida: Academic Press, pp. 26–54.
Roads, C. 1978. "Automated Granular Synthesis of Sound." *Computer Music Journal* 2(2): 61–62. Reprinted as "Granular Synthesis of Sound" in C. Roads and J. Strawn, eds. 1985. *Foundations of Computer Music.* Cambridge, Massachusetts: MIT Press, pp. 145–159.
Rodet, X. 1985. "Time-Domain Formant-Wave-Function Synthesis." *Computer Music Journal* 8(3): 9–14.
Truax, B. 1988. "Real-Time Granular Synthesis with a Digital Signal Processor." *Computer Music Journal* 12(2): 14–26.

Sound Examples

(These are on a soundsheet included with issue 13[1], Spring 1989.) All the examples were realized on the SYTER processor. They are obtained automatically and without any a posteriori corrections.

Example 1. Real speech signal; phrase is played twice.
Example 2. Resynthesis of example 1 without any transformation.
Example 3. Resynthesis of example 1 with frequency transposition effect (ratio = 3).
Example 4. Resynthesis of example 1 with slowdown effect (ratio = 3).
Example 5. Resynthesis of example 1 with acceleration effect (ratio = 2).
Example 6. Real trombone signal.
Example 7. Wavelet cross-synthesis of examples 1 and 6 (modulus of sound 1 and phase of sound 6).
Example 8. Periodic sound (synthetic).
Example 9. Wavelet cross-synthesis of examples 1 and 8.
Example 10. Flutelike real sound.
Example 11. Resynthesis of example 10 without any transformation.
Example 12. Partial resynthesis of example 10, using little dilation parameter voices. This resynthesis permits a high-frequency noise extraction.
Example 13. Partial resynthesis of example 10. The contributions of the voices used for example 12 are eliminated.
Example 14. Real saxophone sound; phrase is played twice.
Example 15. Resynthesis of example 14 without any transformation.
Example 16. Resynthesis of example 14 with frequency transposition effect (ratio = 2).
Example 17. Resynthesis of example 14 with slowdown effect (ratio = 2).
Example 18. Resynthesis of example 14 with brightness effect. The phase modification is given by waveshaping. The distortion function is a sum of the first three Chebyshev polynomials.