MALLAT'S FAST WAVELET ALGORITHM: RECURSIVE COMPUTATION OF CONTINUOUS-TIME WAVELET COEFFICIENTS

Recall discrete-time wavelet series can be computed 2 ways:

1. Direct computation of transform:

average: $x_m(n) = \sum_i x(i) h_0^m(2^m n - i)$

detail: $\mathcal{W}_{2^m} x(i) = \sum_i x(i) h_1^m(2^m n - i)$

Direct computation of inverse transform:

$$x(n) = \sum_{m=1}^{L} \sum_i \mathcal{W}_{2^m} x(i) h_1^m(2^m i - n) + \sum_i x_L(i) h_0^L(2^L i - n)$$

where the filters h_i^M are computed recursively using

$h_0^{m+1}(n) = \sum_i h_0(i) h_0^m(n - 2^m i)$
$h_1^{m+1}(n) = \sum_i h_1(i) h_0^m(n - 2^m i)$

---

2. Recursive computation of transform:

initialization: $x_0(n) = x(n)$
average: $x_m(n) = \sum_i x_{m-1}(i) h_0(2n - i)$
detail: $\mathcal{W}_{2^m} x(n) = \sum_i x_{m-1}(i) h_1(2n - i)$

Note these formulae combine filtering and subsampling.

Recursive computation of inverse transform:

$x_{m-1}(n) = \sum_i h_0(2i - n) x_m(i) + h_1(2i - n) \mathcal{W}_{2^m} x(i)$
Stop at $x_0(n) = x(n)$.

Note time reversal between analysis and synthesis filters.
Discrete-time wavelets implemented by *subband coder*.

Continuous-time wavelet transform is computed directly:

$x(t) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} x_j^i 2^{-i/2} \psi(2^{-i}t - j)$
$x_j^i = \int_{-\infty}^{\infty} x(t) 2^{-i/2} \psi(2^{-i}t - j)$

Can also use scaling function to truncate sum over scale:

$x(t) = \sum_{i=-\infty}^{J} \sum_{j=-\infty}^{\infty} x_j^i 2^{-i/2} \psi(2^{-i}t - j)$
$\quad + \sum_{j=-\infty}^{\infty} \tilde{x}_j^J 2^{-J/2} \phi(2^{-J}t - j)$
$\tilde{x}_j^J = \int_{-\infty}^{\infty} x(t) 2^{-J/2} \phi(2^{-J}t - j)$

Compare to discrete-time direct computation.

Is there any continuous-time counterpart to recursive computation of the discrete-time wavelet series?

There is! The **fast wavelet algorithm** (Mallat 1990)

To link discrete-time and cont-time, recall **2-scale eqns.**

$\phi(t) = \sum_{n=-\infty}^{\infty} g_0(n) 2^{1/2} \phi(2t - n)$
$\psi(t) = \sum_{n=-\infty}^{\infty} g_1(n) 2^{1/2} \phi(2t - n)$

Let $t \to 2^{-i}t - j$, multiply by $2^{-i/2}x(t)$ and $\int_{-\infty}^{\infty} dt$:

$\tilde{x}_j^i = \sum_{n=-\infty}^{\infty} g_0(n) \tilde{x}_{n+2j}^{i-1}; \quad x_j^i = \sum_{n=-\infty}^{\infty} g_1(n) \tilde{x}_{n+2j}^{i-1}$

Since $g_i(n) = h_i(-n), i = 0, 1$ we can rewrite these as

$\tilde{x}_j^i = \sum_{n=-\infty}^{\infty} h_0(n) \tilde{x}_{2j-n}^{i-1} = \sum_{n=-\infty}^{\infty} h_0(2j - n) \tilde{x}_n^{i-1}$
$x_j^i = \sum_{n=-\infty}^{\infty} h_1(n) \tilde{x}_{2j-n}^{i-1} = \sum_{n=-\infty}^{\infty} h_1(2j - n) \tilde{x}_n^{i-1}$

Fastest way to compute cont-time wavelet expansion:

1. Compute $\tilde{x}_j^J$ at *finest* resolution J
2. Recursively compute $\tilde{x}_j^i$ and x_j^i from $\tilde{x}_j^{i-1}$
   Recall bigger $i \to coarser$, so $finer \to coarser$
3. Purely discrete-time processing since wavelet coefficients
4. Implemented using subband coder!