

Examples of Scientific Computing via Wavelets

Daniel Beatty

June 13, 2003

1 Methods of Computing Wavelets

1.1 Convolution

Typically, each component has the form $O \rightarrow (A|D)$ where

- O is the original signal
- A is the average component
- D is the difference component
- $(A|D)$ the signal A concatenated with D .

The general algorithm is

$$\forall i \in [0, M)$$

$$\forall j \in [0, N)$$

$$n=i-j$$

$$\text{if } (n \in [0, M))$$

$$y_i += x_n \cdot h_j$$

1-D Convolution is fairly straight forward from the algorithm.

1.2 2-D Convolution

2-D Convolution is little more complicated, and requires a choice in the manner in which it is computed. For speed, the choice is perform the operation directly on each row and then each column of the matrix.

$\forall i \in \text{rows in the matrix}$

- Generate $S \xrightarrow{\phi} A$ on row i
- Generate $S \xrightarrow{\psi} D$ on row i
- Unite $A \text{ concat } D \rightarrow (A|D)$ in row i of W'

$\forall j \in \text{columns in the matrix}$

- Generate $W' \xrightarrow{\phi} A$ on column j
- Generate $W' \xrightarrow{\psi} D$ on column j
- Unite $A \text{ concat } D \rightarrow (A|D)$ in column j of W'

1.2.1 Chosen Method

The algorithm is as follows for the row tranform (and is similar for the column transform):

$\forall i \in rows$

1. Initialize temporary array/vector to all zeros (x).

2. $\forall k \in columns$

$$(a) \quad \forall l \in ha.Size \quad x += \begin{cases} S_{i,k-l} * hA_l & k-l \in columns \\ 0 & otherwise \end{cases}$$

3. $\forall k \in columns/2$

$result_{i,k} = x_{2k+1}$ (In other words, odd split)

4. Initialize x to all zeros.

5. $\forall k \in columns$

$$(a) \quad \forall l \in hd.Size \quad x += \begin{cases} S_{i,k-l} * hD_l & k-l \in columns \\ 0 & otherwise \end{cases}$$

1.3 1-D Storage Method

The obvious choice is to take either the odd or even indexed values of the average and difference over complete forms, and insert them into the array such that all of the averages are followed by the difference. Each component is subsequently a complete wavelet packet.

1.4 2-D Storage Method

The matrix form below is chosen for its simplicity:

$$B \Rightarrow \begin{matrix} A & V \\ H & D \end{matrix}$$

beginenumerate

6. Average component: produced by filtering the row vectors and the column vectors with the averaging filter.
7. Vertical Component: produced by applying the average filter to the column vectors and the difference filter to the row vectors.
8. Horizontal component: produced by applying the average filter to the row vectors and the difference to the column vectors.
9. Diagonal component: produced by applying the difference filter to both the row and column vectors.

1.5 Multi-Resolution

The wavelet transform (multiresolution) uses private members of the class (hA, hD, xD/yD, xA/yA). Both Haar filters are maintained this way. Also both row and column transforms have average and difference myVector classes for temporary storage. All of these members are allocated and destroyed by the wavelet transform method itself. The simplified algorithm of the row transform is the following:

1. Initialize xA and xD to zero

2. $\forall k \in columns \forall l \in filter$

- $n = k - 1$
- if ($n \in columns$)

$$xA_k = W_{i,n} * hA_l$$

$$xD_k = W_{i,n} * hD_l$$

3. Transfer back to W

$$W_i = xA | xD$$

1. initialize yA and yD to zero

2. $\forall k \in rows \forall l \in filter$

- $n = k - 1$
- if ($n \in columns$)

$$yA_k = W_{i,n} * hA_l$$

$$yD_k = W_{i,n} * hD_l$$

3. Transfer back to W

$$W_j = yA | yD$$

2 Computation Costs

2.1 For 1-D operations

Linear Cost

2.2 2-D Operations

Cost is inherently $O(N^2)$

3 Results

3.1 1-D Transform

Testing of the 1-D wavelet was performed on a sinusoidal wave form of 128 elements. The given function has the equation shown in Figure 1.

$$y[n] = 10 \sin(\frac{n}{128}) - 5 \sin(\frac{n}{64}) + 2 \sin(\frac{3n}{128}) - \sin(\frac{n}{32})$$

Also this produces a functional difference between wavelet inverse transform for odd and even versions. The difference is slight; however, the last element is lost in the even indexed form.

- Odd $R_{2i} = (A_i - D_i)\sqrt{1/2}$ $R_{2i+1} = (A_i + D_i)\sqrt{1/2}$
- Even $R_{2i} = (A_i + D_i)\sqrt{1/2}$ $R_{2i-1} = (A_i - D_i)\sqrt{1/2}$

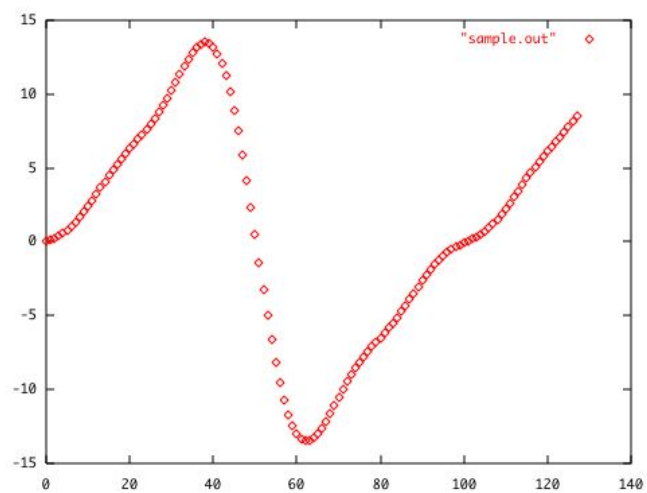


Figure 1: Sample function. The x-axis is the array index (index n). The y value is simple – the value $y[n]$.

3.2 2-D Pictures

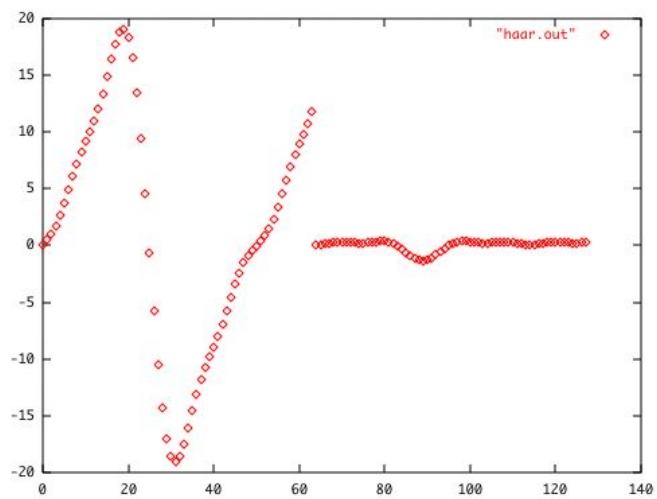


Figure 2: Recovered function. The x-axis is the array index (index n). The y value is simply the value $y[n]$. The function was recovered from an odd indexed wavelet transform.

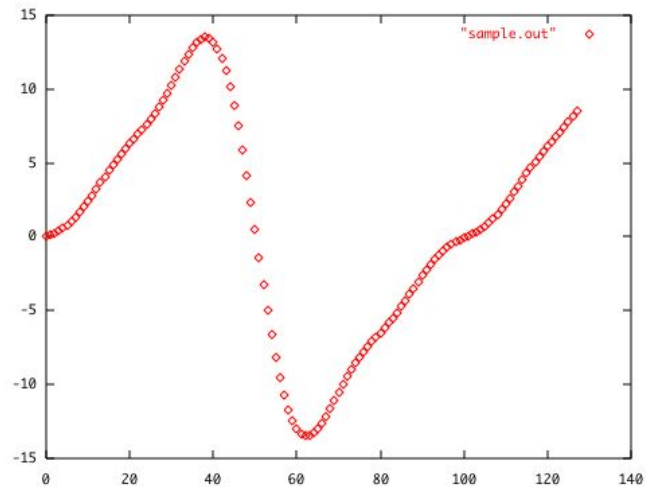


Figure 3: Recovered function. The x-axis is the array index (index n). The y value is simply the value $y[n]$. The function was recovered from an even indexed wavelet transform.



Figure 4: Original Image. This image is the original image.



Figure 5: Recovered Image. This image is the recovered image. Depending on whether the image was saved as a picture first can affect the white spots in the picture. Ringing is also an issue.



Figure 6: Wavelet Transform Image. This image is divided in to average, horizontal, vertical and diagonal components.

[More Pictures](#)



Figure 7: Wavelet Transform Image. This image is divided in to average, horizontal, vertical and diagonal components, using the vector-matrix version.

[More pictures](#)



Figure 8: Recovered Image (Vector-Matrix Method). This image is the recovered image. This version avoids the ringing by using the vector-matrix version which is more aligned for the inverse wavelet transform.



Figure 9: Wavelet Transform Image. This image is divided in to average, horizontal, vertical and diagonal components using multi-resolution wavelet transform. Note the the average component was transformed one step further.

4 Matrix Multiplication

4.1 Simple Proof

A reminder from Dr. Sinzinger, wavelets and their inverses are linear operators. As he correctly reminded me any linear operator has the following properties:

1. $L(AB) = L(A)\dot{L}(B)$
2. ψ^{-1} is a linear operator
3. $\psi^{-1}(\psi(A)\dot{\psi}(B)) = \psi^{-1}(\psi(A))\dot{\psi}^{-1}(\psi(B))$
4. $\psi^{-1}(\psi(A)) = A$
5. $\psi^{-1}(\psi(B)) = B$
6. Therefore: $AB = \psi^{-1}(\psi(A)\dot{\psi}(B))$

Thus this is sufficient proof that wavelet matrix multiplication is sound.

4.2 Limits spelled out by the proof

One catch-22 to be had with the simple proof. The linear transform has to be the same linear transform on both matrices. If the matrices are the same size, $(N \times N)$ then this is no-brainer. If they are $M \times P$ and $P \times N$, then it is not clear that the wavelet transforms of each are the same operator.

4.3 Chain Link Structure

Class Chain Link members: length hooks

Class Hook members: length l2norm links

Class Link members: id value

5 Items to come

5.1 Matrix Inversion

5.2 PDE's in Wavelet Space