20 years of Wavelets : Tales from Industry and Academia

DANIEL LEMIRE

Research Officer, NRC

Adjunct Professor, UNB

http ://www.ondelette.com/acadia/

20 years of Wavelets : Tales from Industry and Academia

DANIEL LEMIRE

Research Officer, NRC

Adjunct Professor, UNB

http ://www.ondelette.com/acadia/

Motivation : why care about wavelets ?

– Neat mathematical beasts

Motivation : why care about wavelets ?

– Neat mathematical beasts
– Tremendously useful in industry

Motivation : why care about wavelets ?

- Neat mathematical beasts
- Tremendously useful in industry
- Think : FBI Fingerprint, JPEG2000, DSP

Motivation : why care about wavelets ?

– Neat mathematical beasts
– Tremendously useful in industry
– Think : FBI Fingerprint, JPEG2000, DSP
– Nice mix of industry-academia work to achieve current result

Goal of this talk

– Justify wavelets from earlier approaches

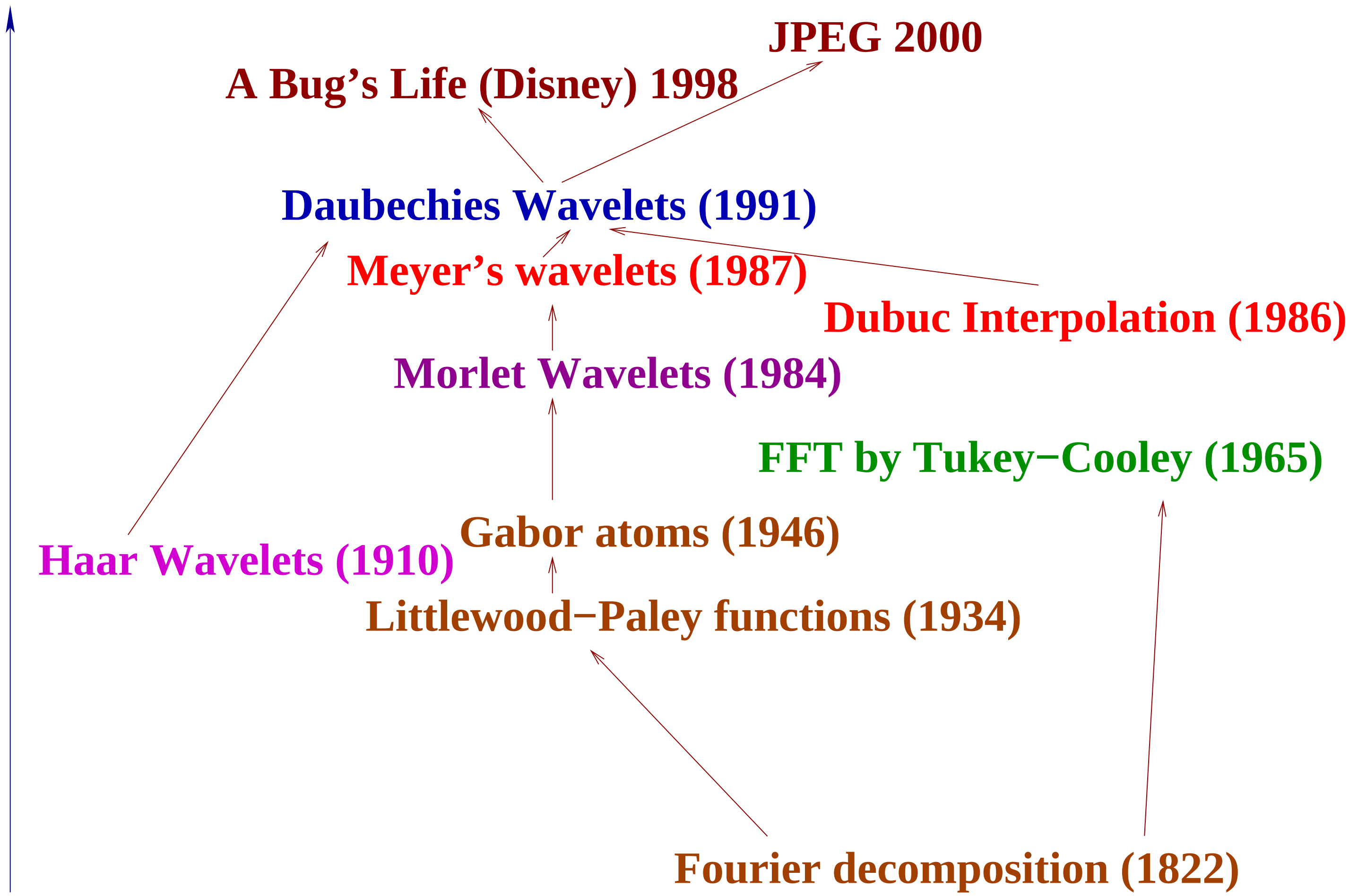
Goal of this talk

- Justify wavelets from earlier approaches
- Fully derive Daubechies wavelets using elementary math !

Goal of this talk

– Justify wavelets from earlier approaches
– Fully derive Daubechies wavelets using elementary math !
– Wavelets **don't** have to be hard !

Approximative and Incomplete Timeline



Remember Joseph

– Fourier Transform *convolves* signal with $\cos(fx)$

Remember Joseph

- Fourier Transform *convolves* signal with $\cos(fx)$
- Implemented in $O(n \log n)$ with FFT

Remember Joseph

- Fourier Transform *convolves* signal with $\cos(fx)$
- Implemented in $O(n \log n)$ with FFT
- everybody uses it every day

Remember Joseph

- Fourier Transform *convolves* signal with $\cos(fx)$
- Implemented in $O(n \log n)$ with FFT
- everybody uses it every day
- Example : JPG, cell phones...

Remember Joseph

- Fourier Transform *convolves* signal with $\cos(fx)$
- Implemented in $O(n \log n)$ with FFT
- everybody uses it every day
- Example : JPG, cell phones...
- Uses bunch of **orthonormal functions**

Remember Joseph

- Fourier Transform *convolves* signal with $\cos(fx)$
- Implemented in $O(n \log n)$ with FFT
- everybody uses it every day
- Example : JPG, cell phones...
- Uses bunch of **orthonormal functions**
- Wait !

Remember Joseph

- Fourier Transform *convolves* signal with $\cos(fx)$
- Implemented in $O(n \log n)$ with FFT
- everybody uses it every day
- Example : JPG, cell phones...
- Uses bunch of **orthonormal functions**
- Wait ! What does orthonormal means ?

Orthogonal Transforms and Why They are Good

– T is orthonormal if $\langle x, y \rangle =$

Orthogonal Transforms and Why They are Good

– T is orthonormal if $\langle x, y \rangle = \langle T(x), T(y) \rangle$

Orthogonal Transforms and Why They are Good

– T is orthonormal if $\langle x, y \rangle = \langle T(x), T(y) \rangle$
– Linear algebra : $T^T T =$

Orthogonal Transforms and Why They are Good

- T is orthonormal if $\langle x, y \rangle = \langle T(x), T(y) \rangle$
- Linear algebra : $T^T T = I$

Orthogonal Transforms and Why They are Good

- T is orthonormal if $\langle x, y \rangle = \langle T(x), T(y) \rangle$
- Linear algebra : $T^T T = I$
- Given a signal x,

Orthogonal Transforms and Why They are Good

- T is orthonormal if $\langle x, y \rangle = \langle T(x), T(y) \rangle$
- Linear algebra : $T^T T = I$
- Given a signal x, $\langle x, x \rangle$ is its energy

Orthogonal Transforms and Why They are Good

- T is orthonormal if $\langle x, y \rangle = \langle T(x), T(y) \rangle$
- Linear algebra : $T^T T = I$
- Given a signal x, $\langle x, x \rangle$ is its energy
- Orthonormality $\Rightarrow$ *preserves* the energy !

Orthogonality by Example

– Look at a signal $\{0.8, 3.8, 4.6, 1.5\}$

Orthogonality by Example

- Look at a signal $\{0.8, 3.8, 4.6, 1.5\}$
- FFT is $\{10.7, -3.8-2.3j, 0.1, -3.8+2.3j\}$

Orthogonality by Example

- Look at a signal $\{0.8, 3.8, 4.6, 1.5\}$
- FFT is $\{10.7, -3.8-2.3j, 0.1, -3.8+2.3j\}$
- What happens if I replace it by $\{10.7, -3.8-2.3j, 0.0, -3.8+2.3j\}$?

Orthogonality by Example

- Look at a signal $\{0.8, 3.8, 4.6, 1.5\}$
- FFT is $\{10.7, -3.8-2.3j, 0.1, -3.8+2.3j\}$
- What happens if I replace it by $\{10.7, -3.8-2.3j, 0.0, -3.8+2.3j\}$?
- The energy will go down by exactly 0.1^2

Orthogonality by Example

- Look at a signal $\{0.8, 3.8, 4.6, 1.5\}$
- FFT is $\{10.7, -3.8-2.3j, 0.1, -3.8+2.3j\}$
- What happens if I replace it by $\{10.7, -3.8-2.3j, 0.0, -3.8+2.3j\}$?
- The energy will go down by exactly 0.1^2
- means we can remove small coefficients without too much impact

Orthogonality by Example

- Look at a signal $\{0.8, 3.8, 4.6, 1.5\}$
- FFT is $\{10.7, -3.8-2.3j, 0.1, -3.8+2.3j\}$
- What happens if I replace it by $\{10.7, -3.8-2.3j, 0.0, -3.8+2.3j\}$?
- The energy will go down by exactly 0.1^2
- means we can remove small coefficients without too much impact
- Inverse is :$\{0.775, 3.825, 4.575, 1.525\}$

Other reasons to like orthonormality

– Suppose x and y are uncorrelated signals

Other reasons to like orthonormality

- Suppose x and y are uncorrelated signals
- Means that $\langle x, y \rangle$ is zero

Other reasons to like orthonormality

- Suppose x and y are uncorrelated signals
- Means that $\langle x, y \rangle$ is zero
- You have the transforms are also uncorrelated

Other reasons to like orthonormality

- Suppose x and y are uncorrelated signals
- Means that $\langle x, y \rangle$ is zero
- You have the transforms are also uncorrelated
- $\langle T(x), T(y) \rangle$ is zero

Other reasons to like orthonormality

- Suppose x and y are uncorrelated signals
- Means that $\langle x, y \rangle$ is zero
- You have the transforms are also uncorrelated
- $\langle T(x), T(y) \rangle$ is zero
- $T(x)$, $T(y)$ uncorrelated

Other reasons to like orthonormality

- Suppose x and y are uncorrelated signals
- Means that $\langle x, y \rangle$ is zero
- You have the transforms are also uncorrelated
- $\langle T(x), T(y) \rangle$ is zero
- $T(x)$, $T(y)$ uncorrelated
- IOW : Orthogonality means things are sane

Summary : we like orthonormality because...

- Can be inverted (ex. Fourier) and inverse is also orthonormal
- Preserves energy : big things go to big things, small things go to small things

Summary : we like orthonormality because...

- Can be inverted (ex. Fourier) and inverse is also orthonormal
- Preserves energy : big things go to big things, small things go to small things
- Preserves correlation or *uncorrelation* : unrelated things remain unrelated, related things remain re-lated

Summary : we like orthonormality because...

- Can be inverted (ex. Fourier) and inverse is also orthonormal
- Preserves energy : big things go to big things, small things go to small things
- Preserves correlation or *uncorrelation* : unrelated things remain unrelated, related things remain related

Why do you want to transform a signal

– Original idea : simplify

Why do you want to transform a signal

- Original idea : simplify
- Example : $v_1 = \{1,2,3\}$,$v_2 = \{2,4,6\}$, $v_3 = \{3,6,9\}$

Why do you want to transform a signal

- Original idea : simplify
- Example : $v_1 = \{1,2,3\}$,$v_2 = \{2,4,6\}$, $v_3 = \{3,6,9\}$
- Rotate axis : $v_1 = \{1,0,0\}$,$v_2 = \{2,0,0\}$, $v_3 = \{3,0,0\}$

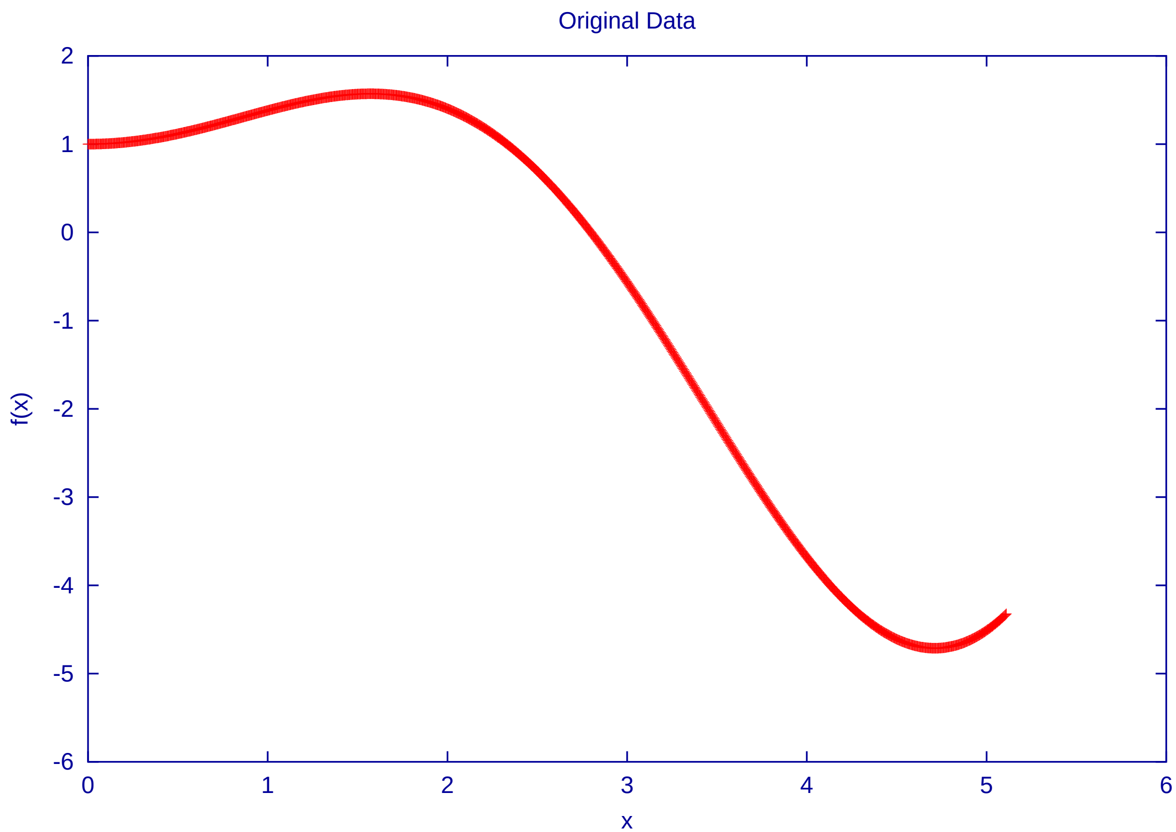
Why do you want to transform a signal

- Original idea : simplify
- Example : $v_1 = \{1,2,3\}$,$v_2 = \{2,4,6\}$, $v_3 = \{3,6,9\}$
- Rotate axis : $v_1 = \{1,0,0\}$,$v_2 = \{2,0,0\}$, $v_3 = \{3,0,0\}$
- *Simple* often means **lots of zeros !**

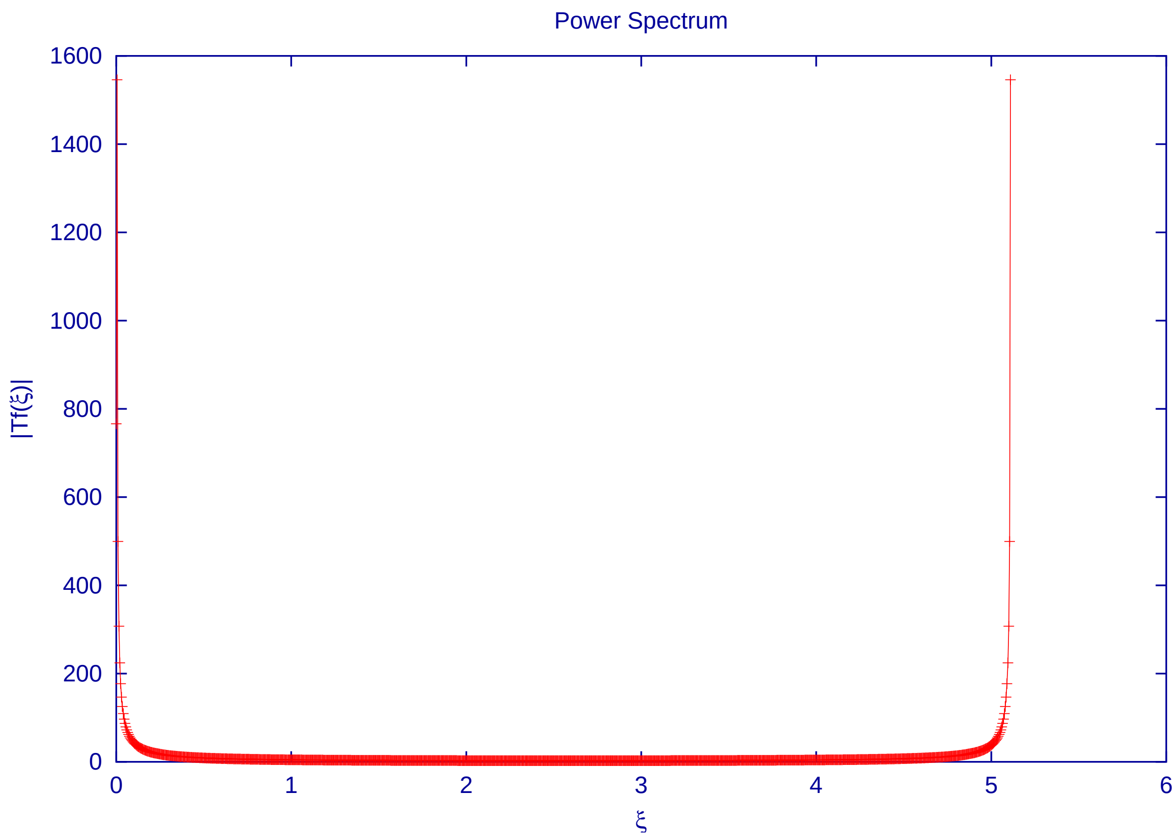
Why do you want to transform a signal

– Original idea : simplify
– Example : $v_1 = \{1,2,3\}$,$v_2 = \{2,4,6\}$, $v_3 = \{3,6,9\}$
– Rotate axis : $v_1 = \{1,0,0\}$,$v_2 = \{2,0,0\}$, $v_3 = \{3,0,0\}$
– *Simple* often means **lots of zeros !**
– Result : FFT (Power Spectrum) of smooth signal has lots of zeros !

Why Transform ? (an example)



Why Transform ? (result from example)



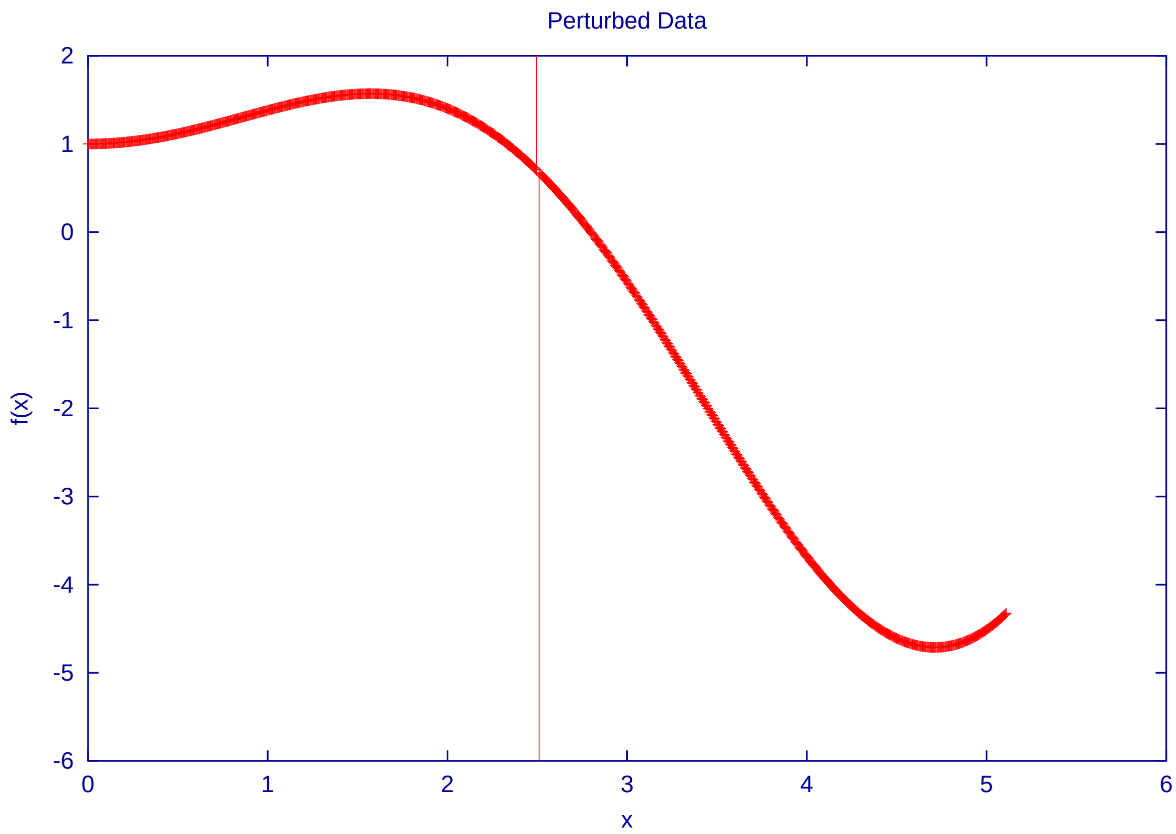
What’s wrong with Fourier ?

– It uses cosines... Cosines are not local !

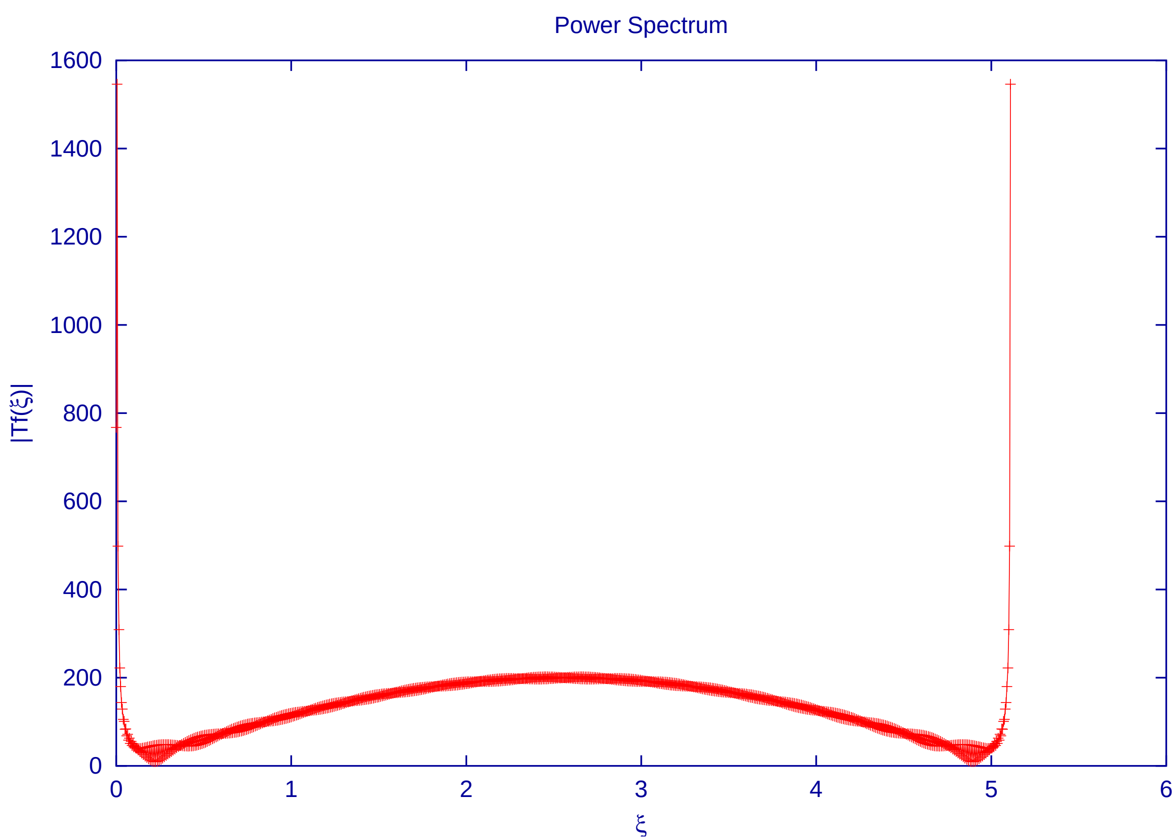
What’s wrong with Fourier ?

- It uses cosines... Cosines are not local !
- Lack of locality is a **big**, fundamental flaw !

Why Locality ? (a spike in the data)



Why Locality ? (a very different spectrum)



Can’t you fix Fourier ?

– You can use *sliding windows*

Can’t you fix Fourier ?

– You can use *sliding windows*

– Given a signal, what the best size for the window ?

Can’t you fix Fourier ?

– You can use *sliding windows*
– Given a signal, what the best size for the window ?
– Window size = time localization needed

Can’t you fix Fourier ?

- You can use *sliding windows*
- Given a signal, what the best size for the window ?
- Window size = time localization needed
- Lots of events : use short window (low freq resolution)

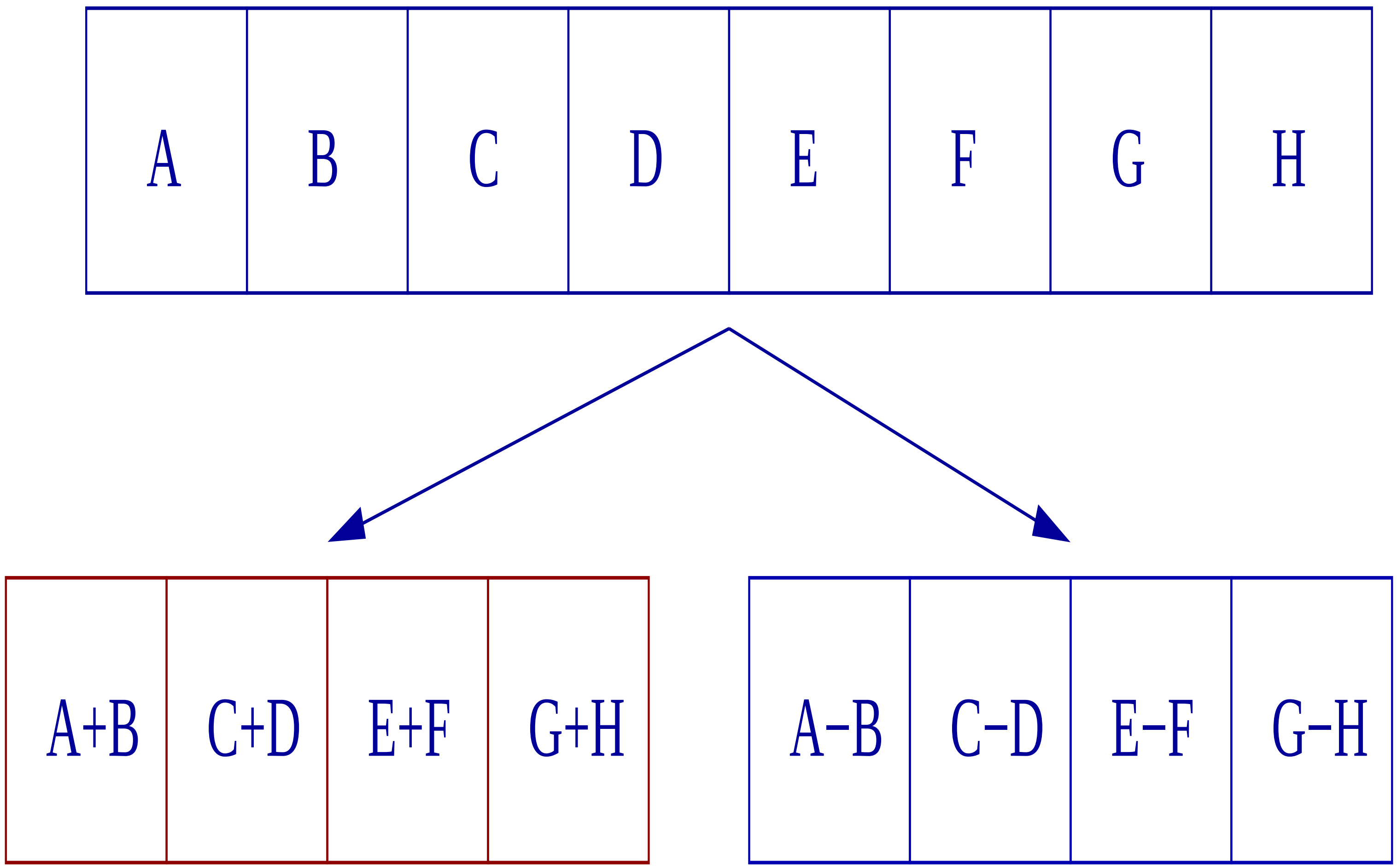
Can’t you fix Fourier ?

- You can use *sliding windows*
- Given a signal, what the best size for the window ?
- Window size = time localization needed
- Lots of events : use short window (low freq resolution)
- Few events : use large window (better freq resolution)

Can’t you fix Fourier ?

- You can use *sliding windows*
- Given a signal, what the best size for the window ?
- Window size = time localization needed
- Lots of events : use short window (low freq resolution)
- Few events : use large window (better freq resolution)
- Suppose you want to track both (think Doppler signal) ?

A simple alternative : Haar (1910)



Why is Haar Nice ?

– It is **orthonormal** : preserves energy and correlation (when normalized by $1/sqrt(2)$)

Why is Haar Nice ?

- It is **orthonormal** : preserves energy and correlation (when normalized by $1/sqrt(2)$)
- It is perfectly invertible

Why is Haar Nice ?

- It is **orthonormal** : preserves energy and correlation (when normalized by $1/sqrt(2)$)
- It is perfectly invertible
- A smooth signal will have a lot of **zeros** : $x_i - x_{i+1}$ is small when sampling is high and signal is smooth

Why is Haar Nice ?

- It is **orthonormal** : preserves energy and correlation (when normalized by $1/sqrt(2)$)
- It is perfectly invertible
- A smooth signal will have a lot of **zeros** : $x_i - x_{i+1}$ is small when sampling is high and signal is smooth
- Example : $1, 1.2, 1.3, 1.2$ transforms to $-0.2, -0.1$ and $2.2, 2.5$.

Is one scale enough

– If signal is nearly constant, only half the coefs will be ≈ 0

Is one scale enough

- If signal is nearly constant, only half the coefs will be ≈ 0
- Can we improve ? ? ?

Is one scale enough

- If signal is nearly constant, only half the coefs will be ≈ 0
- Can we improve ? ? ?
- Yes !

Is one scale enough

- If signal is nearly constant, only half the coefs will be ≈ 0
- Can we improve???
- Yes! Just repeat!

Is one scale enough

- If signal is nearly constant, only half the coefs will be ≈ 0
- Can we improve ? ? ?
- Yes ! Just repeat !
- $1, 1.2, 1.3, 1.2 \rightarrow -0.2, -0.1$ and $2.2, 2.5$.

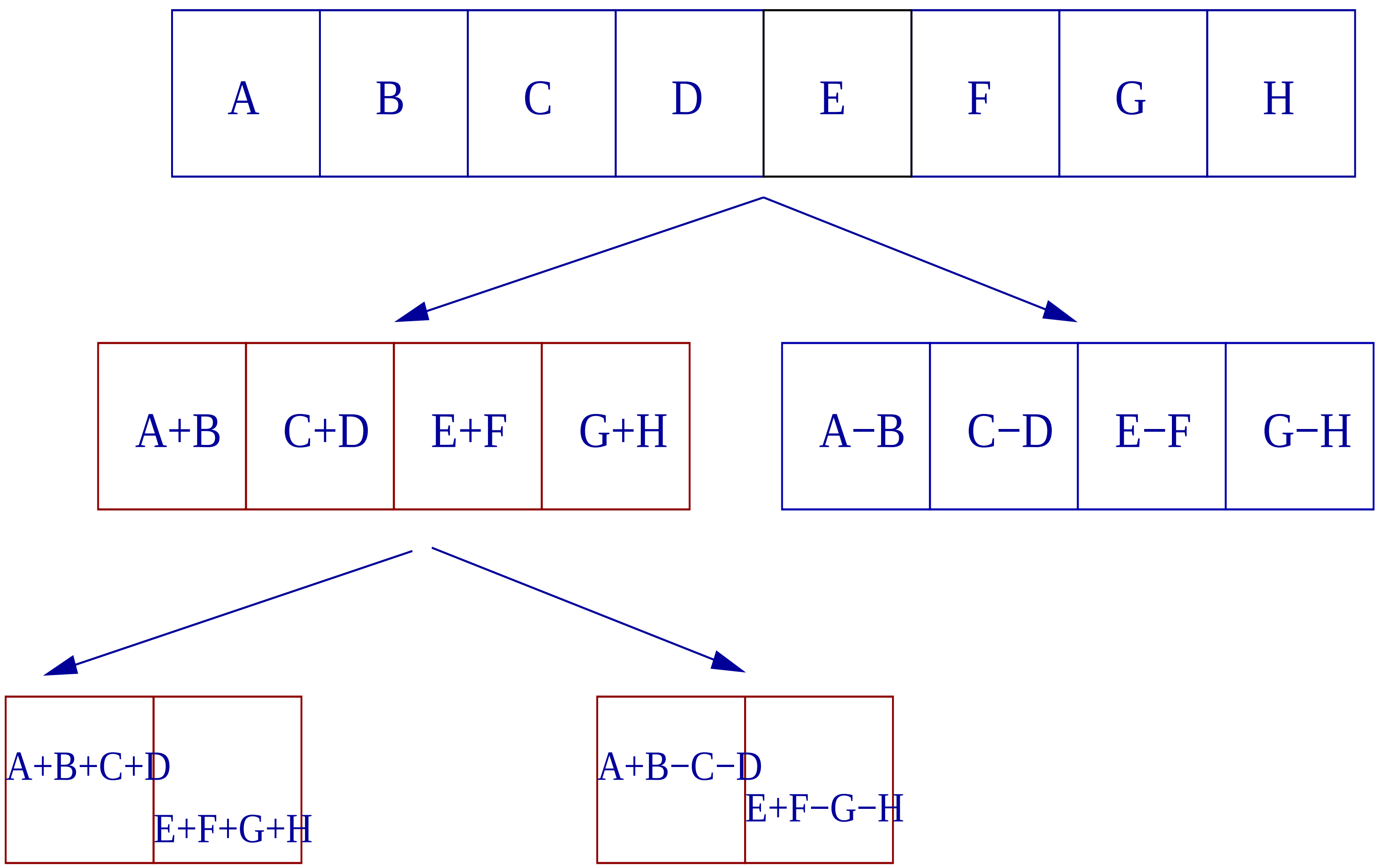
Is one scale enough

- If signal is nearly constant, only half the coefs will be ≈ 0
- Can we improve ? ? ?
- Yes ! Just repeat !
- $1, 1.2, 1.3, 1.2 \longrightarrow -0.2, -0.1$ and $2.2, 2.5$.
- $2.2, 2.5 \longrightarrow -0.3$ and 4.7

Is one scale enough

- If signal is nearly constant, only half the coefs will be ≈ 0
- Can we improve???
- Yes! Just repeat!
- $1, 1.2, 1.3, 1.2 \longrightarrow -0.2, -0.1$ and $2.2, 2.5$.
- $2.2, 2.5 \longrightarrow -0.3$ and 4.7
- This is magical : everything goes to near zero!!!

Two-scale Haar



Going multiscale

– Let’s think about what we just did...

Going multiscale

– Let’s think about what we just did...
– We applied a transform...

Going multiscale

- Let’s think about what we just did...
- We applied a transform...
- Then reapplied the transform to the output of the transform

Going multiscale

- Let’s think about what we just did...
- We applied a transform...
- Then reapplied the transform to the output of the transform
- And then we could apply the transform to the output of the output of the ouput...

Going multiscale

- Let's think about what we just did...
- We applied a transform...
- Then reapplied the transform to the output of the transform
- And then we could apply the transform to the output of the output of the output of the ouput...
- Kind of like a

Going multiscale

- Let's think about what we just did...
- We applied a transform...
- Then reapplied the transform to the output of the transform
- And then we could apply the transform to the output of the output of the output of the ouput...
- Kind of like a **mathematical zooming out**!

Going multiscale

- Let's think about what we just did...
- We applied a transform...
- Then reapplied the transform to the output of the transform
- And then we could apply the transform to the output of the output of the ouput...
- Kind of like a **mathematical zooming out**!
- This leads to self-similarity (fractals) and multiscale analysis...

Going further : Daubechies (1991)

– A French engineer (Morlet) *invented* wavelets, and the math was done by Meyer

Going further : Daubechies (1991)

– A French engineer (Morlet) *invented* wavelets, and the math was done by Meyer
– However, wavelets really took off thanks to **Ingrid Daubechies**

Going further : Daubechies (1991)

- A French engineer (Morlet) *invented* wavelets, and the math was done by Meyer
- However, wavelets really took off thanks to **Ingrid Daubechies**
- Her work can be seen as a generalization of Haar whereas Morlet's wavelets were not

What's wrong with Haar ?

- Suppose signal is linear $f(x) = ax + b$ with $a \neq 0$, then you'd want lots of zeros...

What's wrong with Haar ?

– Suppose signal is linear $f(x) = ax + b$ with $a \neq 0$, then you'd want lots of zeros...
– Not so with Haar... $1,2,3,4,5,6 \rightarrow -1,-1,-1, 3,7,11$

What's wrong with Haar ?

- Suppose signal is linear $f(x) = ax + b$ with $a \neq 0$, then you'd want lots of zeros...
- Not so with Haar... $1,2,3,4,5,6 \rightarrow -1,-1,-1, 3,7,11$
- Needs really high sampling frequency

What's wrong with Haar ?

- Suppose signal is linear $f(x) = ax + b$ with $a \neq 0$, then you'd want lots of zeros...
- Not so with Haar... $1, 2, 3, 4, 5, 6 \rightarrow -1, -1, -1, 3, 7, 11$
- Needs really high sampling frequency
- Not very realistic

Adding two terms

– So, Haar is of the form $ax_i + bx_{i+1}$ and $cx_i + dx_{i+1}$ where $a = 1, b = 1, c = 1, d = -1$ (plus normalization...)

Adding two terms

– So, Haar is of the form $ax_i + bx_{i+1}$ and $cx_i + dx_{i+1}$ where $a = 1, b = 1, c = 1, d = -1$ (plus normalization...)

– Must satisfy $c + d = 0$ (constant goes to zero)

Adding two terms

- So, Haar is of the form $ax_i + bx_{i+1}$ and $cx_i + dx_{i+1}$ where $a = 1, b = 1, c = 1, d = -1$ (plus normalization...)
- Must satisfy $c + d = 0$ (constant goes to zero)
- $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is orthonormal implies...
- $ac + bd = 0$, $a^2 + b^2 = 1$, and $c^2 + d^2 = 1$

Setting up the equations

– So let's look at things of the form $a_0 x_i + a_1 x_{i+1} + a_2 x_{i+2} + a_3 x_{i+3}$

Setting up the equations

- So let's look at things of the form $a_0 x_i + a_1 x_{i+1} + a_2 x_{i+2} + a_3 x_{i+3}$
- Set $a_0 + a_1 + a_2 + a_3 + a_4 = 0$ (constant signal goes to zero)

Setting up the equations

- So let's look at things of the form $a_0x_i + a_1x_{i+1} + a_2x_{i+2} + a_3x_{i+3}$
- Set $a_0 + a_1 + a_2 + a_3 + a_4 = 0$ (constant signal goes to zero)
- Set $a_0 \times 0 + a_1 \times 1 + a_2 \times 2 + a_3 \times 3 = 0$ (linear slope goes to zero)

Setting up the equations

- So let's look at things of the form $a_0x_i + a_1x_{i+1} + a_2x_{i+2} + a_3x_{i+3}$
- Set $a_0 + a_1 + a_2 + a_3 + a_4 = 0$ (constant signal goes to zero)
- Set $a_0 \times 0 + a_1 \times 1 + a_2 \times 2 + a_3 \times 3 = 0$ (linear slope goes to zero)
- We want orthonormality which implies, just like with Haar,
- Set $a_0a_2 + a_1a_3 = 0$ (orthogonality condition)

Using Maxima (free software)

```
(C1) solve([a_1+2*a_2+3*a_3=0,a_0+a_1+a_2+a_3=0],[a_
                A2 + 3 a_0          A2 - a_0
(D1) [[A1 = - ---------, A3 = - -------]]
                   2                 2
(C2) ratsimp(expand(subst(D1,a_0*a_2+a_1*a_3)));
          2                 2
        A2  + 6 A0 A2 - 3 A0
(D2)  ---------------------
                4
(C3) solve(%=0,A2);
(D3) [A2=(- 2 SQRT(3) - 3)a_0, A2=(2 SQRT(3)-3)a_0]
```

Solution...

$$[a_0, a_1, a_2, a_3] =$$

$$\left[a_0, \frac{-\left(2\sqrt{3}-3\right)a_0 - 3\,a_0}{2}, \left(2\sqrt{3}-3\right)a_0, \frac{a_0 - \left(2\sqrt{3}-3\right)a_0}{2}\right]$$

Solution...

$$[a_0, a_1, a_2, a_3] =$$

$$\left[a_0, \frac{-\left(2\sqrt{3}-3\right)a_0 - 3a_0}{2}, \left(2\sqrt{3}-3\right)a_0, \frac{a_0 - \left(2\sqrt{3}-3\right)a_0}{2}\right]$$

Notice how everything on the right depends linearly on a_0 ?

Finishing...

– Remains to set a_0, we require that $a_0^2 + a_1^2 + a_2^2 + a_3^3 = 1$ (conservation of energy)

Finishing...

- Remains to set a_0, we require that $a_0^2 + a_1^2 + a_2^2 + a_3^3 = 1$ (conservation of energy)
- Get $a_0 = 0.4829, a_1 = -0.8365, a_2 = 0.2241, a_3 = 0.1294$

Finishing...

– Remains to set a_0, we require that $a_0^2 + a_1^2 + a_2^2 + a_3^3 = 1$ (conservation of energy)
– Get $a_0 = 0.4829, a_1 = -0.8365, a_2 = 0.2241, a_3 = 0.1294$
– We've fully derived Daubechies 2 wavelet (can go much higher)

Finishing...

- Remains to set a_0, we require that $a_0^2+a_1^2+a_2^2+a_3^3=1$ (conservation of energy)
- Get $a_0=0.4829, a_1=-0.8365, a_2=0.2241, a_3=0.1294$
- We've fully derived Daubechies 2 wavelet (can go much higher)
- Can use Cohen's formula to derive lowpass filter : $-a_3, a_2, -a_1, a_0$ (this is orthogonal to a_0, a_1, a_2, a_3)

What we just did

– use 4 conditions : constant signal goes to zero, linear slope goes to zero, and orthonormality

What we just did

- use 4 conditions : constant signal goes to zero, linear slope goes to zero, and orthonormality
- enough to derive wavelet filters up to normalization

What we just did

- use 4 conditions : constant signal goes to zero, linear slope goes to zero, and orthonormality
- enough to derive wavelet filters up to normalization
- get normalization by requiring conservation of energy

What we just did

- use 4 conditions : constant signal goes to zero, linear slope goes to zero, and orthonormality
- enough to derive wavelet filters up to normalization
- get normalization by requiring conservation of energy
- That is all !

So ?

– We’ve fully derived the Daubechies wavelets !

So ?

– We’ve fully derived the Daubechies wavelets !
– Anyone is crying ?

Daubechies needs multiscale too !

– For the same reason Haar needed to go multiscale...

Daubechies needs multiscale too !

- For the same reason Haar needed to go multiscale...
- Daubechies needs to be multiscale

Daubechies needs multiscale too !

– For the same reason Haar needed to go multiscale...
– Daubechies needs to be multiscale
– So we get more zeroes !

Possible Applications

– unassuming denoising :

Possible Applications

- unassuming denoising : wavelet shrinkage, much better than lowpass because doesn’t kill specific frequencies

Possible Applications

– unassuming denoising : wavelet shrinkage, much better than lowpass because doesn’t kill specific frequencies
– compression :

Possible Applications

– unassuming denoising : wavelet shrinkage, much better than lowpass because doesn’t kill specific frequencies
– compression : embedded zerotrees,

Possible Applications

– unassuming denoising : wavelet shrinkage, much better than lowpass because doesn't kill specific frequencies
– compression : embedded zerotrees, JPEG2000,

Possible Applications

- unassuming denoising : wavelet shrinkage, much better than lowpass because doesn't kill specific frequencies
- compression : embedded zerotrees, JPEG2000, FBI, better than Fourier because localization buil-tin

Are Wavelets better than Fourier ?

– No. Fourier has perfect frequency resolution,

Are Wavelets better than Fourier ?

– No. Fourier has perfect frequency resolution, but no time resolution

Are Wavelets better than Fourier ?

- No. Fourier has perfect frequency resolution, but no time resolution
- Windowed Fourier has fixed time resoluton, and a decreased frequency resolution

Are Wavelets better than Fourier ?

- No. Fourier has perfect frequency resolution, but no time resolution
- Windowed Fourier has fixed time resoluton, and a decreased frequency resolution
- Wavelets have various frequency/time resolu-tions :

Are Wavelets better than Fourier ?

– No. Fourier has perfect frequency resolution, but no time resolution
– Windowed Fourier has fixed time resoluton, and a decreased frequency resolution
– Wavelets have various frequency/time resolutions : high time resolution at high freq. and high freq. resoluton at low freq.

Are Wavelets better than Fourier ?

- No. Fourier has perfect frequency resolution, but no time resolution
- Windowed Fourier has fixed time resoluton, and a decreased frequency resolution
- Wavelets have various frequency/time resolutions : high time resolution at high freq. and high freq. resoluton at low freq.
- We have to live with uncertainty principle :

Are Wavelets better than Fourier ?

- No. Fourier has perfect frequency resolution, but no time resolution
- Windowed Fourier has fixed time resoluton, and a decreased frequency resolution
- Wavelets have various frequency/time resolutions : high time resolution at high freq. and high freq. resoluton at low freq.
- We have to live with uncertainty principle : can't have both the frequency and the position !

What is the best wavelet ?

– As you increase the number of coefficients (taps),

What is the best wavelet ?

– As you increase the number of coefficients (taps), you get more zeroes

What is the best wavelet ?

– As you increase the number of coefficients (taps), you get more zeroes
– In real life, beyond 16 coefficients, there is no more gains...

What is the best wavelet ?

- As you increase the number of coefficients (taps), you get more zeroes
- In real life, beyond 16 coefficients, there is no more gains...
- Haar is still useful for some signals though

What is the best wavelet ?

- As you increase the number of coefficients (taps), you get more zeroes
- In real life, beyond 16 coefficients, there is no more gains...
- Haar is still useful for some signals though
- So this is an application specific question.

What is the best wavelet ?

- As you increase the number of coefficients (taps), you get more zeroes
- In real life, beyond 16 coefficients, there is no more gains...
- Haar is still useful for some signals though
- So this is an application specific question.
- Likely not to be a critical question in practice.

What about 2D, images ?

- just apply transform on rows then to columns
- It is possible to use **non-separable** wavelets

What about 2D, images ?

- just apply transform on rows then to columns
- It is possible to use **non-separable** wavelets
- Not much use though

Conclusion

– We argued that transformations giving out lots of zeroes were good

Conclusion

- We argued that transformations giving out lots of zeroes were good
- We argued in favor of orthonormality

Conclusion

- We argued that transformations giving out lots of zeroes were good
- We argued in favor of orthonormality
- From this, using grade 12 algebra, we derived Daubechies wavelets

Conclusion

- We argued that transformations giving out lots of zeroes were good
- We argued in favor of orthonormality
- From this, using grade 12 algebra, we derived Daubechies wavelets
- Same stuff that's used in JPEG2000 and other wavelet software

Question 1

Can you really compute the highpass filter on their own ?

Yep. You can worry about the lowpass filters only at the very end (Cohen’s formula).

Question 2

You say that orthogonality implies $a_0a_2 + a_1a_3 = 0$*, where does that come from ?*

We require the following vectors to be orthogonal...

$$(a_0, a_1, a_2, a_3, 0, 0)(0, 0, a_0, a_1, a_2, a_3)$$

Question 3

You say that energy conservation implies $a_0^2+a_1^2+a_2^2+a_3^2=1$*, shouldn't that be* $a_0^2+a_1^2=1$ *and* $a_2^2+a_3^2=1$*?*

Nope. We require these vectors to have unit norm...

$$(a_0, a_1, a_2, a_3, 0, 0)(0, 0, a_0, a_1, a_2, a_3)$$

Question 4

What's Cohen's formula ?

Simply put, it is the observation that these vectors are orthogonal...

$$(a_0, a_1, a_2, a_3)(-a_3, a_2, -a_1, a_0)$$

Question 4

How do you generalize to the case where you have 8 coefficients to solve for ?

Do the same kind of computations, but with these vectors (must all be orthonormal)

$$(\quad a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a8, 0, 0, 0, 0, 0, 0 \quad)$$
$$(\quad 0, 0, a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, 0, 0, 0, 0 \quad)$$
$$(\quad 0, 0, 0, 0, a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, 0, 0 \quad)$$
$$(\quad 0, 0, 0, 0, 0, 0, a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7 \quad)$$

Rest of the answer left as an exercise.