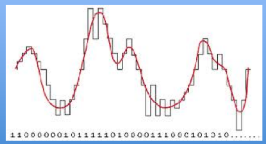
Image Data Compression

Wavelet transform

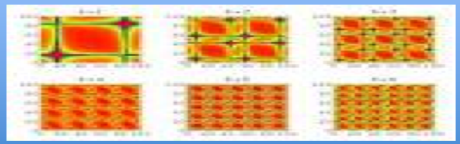
Image signal path: overview

Data acquisition

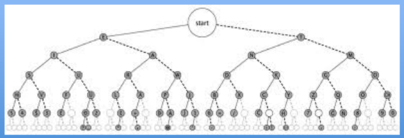
Data reduction (Quantization)



De-correlation (Transform+Quantization)



Coding (Pre-coding + Entropy coding)



Output data:
Continuous multi-dimensional analog signal

Compression:
Physical sensor limitations, DAQ settings

Output data:
Correlated fully digital multi-dimensional symbol stream

Compression:
Spatio-temporal quantization, value quantization

Output data:
Linear stream of de-correlated symbols

Compression:
Truncation of irrelevant signal parts, value quantization

Output data:
Linear stream of bits

Compression:
Reversible suppression of the redundant information

Channel

...

[Some incomplete] classification of de-correlation methods

De-correlation

- **Signal prediction**
  - N-th order Markov process (1D) or multi-dimensional Markov field to reduce entropy of prediction error
  - Linear / non-linear, fixed / adaptive predictors
- **Signal transform**
  - **Linear transforms**
    - Discrete linear decomposition of 2D signals over basis functions in order to concentrate energy
    - Wavelet-type: DFT, DCT, WHT, DWT
    - Filter banks
    - Data-based: KLT
  - **Non-Linear transforms**
    - Fractal transform

Reminder: generic discrete transforms

Input discrete signal: $x[n], n = 0, \dots, N - 1$ **Matrix notation:** an N-dimensional vector $\vec{x}$

Direct transformation: $X[k] = \sum_n a[k,n] \cdot x[n]$ **Matrix notation:** $\vec{X} = A \cdot \vec{x}$

Inverse transformation: $x[n] = \sum_k b[n,k] \cdot X[n]$ **Matrix notation:** $\vec{x} = B \cdot \vec{X}$,

"Condition of perfect reconstruction": $B = A^{-1}$

Basis vectors (functions): $\vec{b}_k = (b[0,k], \dots, b[N-1,k])^T$ **Matrix notation:** k-th column of B

A transform decomposes a signal as a weighted sum of basis functions: $\vec{x} = \sum_k \vec{b}_k \cdot X[k]$

Important types of linear transforms:

Orthogonal: $\sum_k a[k,i] \cdot a[k,j] = C_A \cdot \delta_{ij}$ **Matrix notation:** $A^T \cdot A = C_A \cdot I$

Orthonormal: orthogonal and $C_A = 1$ **Matrix notation:** $A^T \cdot A = I$, i.e. $B = A^T$

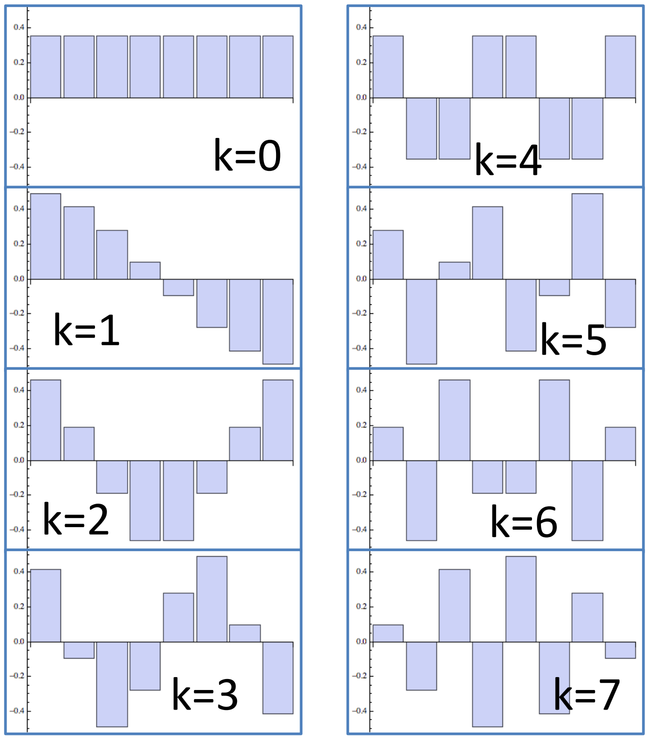
Bi-orthogonal: $\sum_k a[k,i] \cdot b[k,j] = C \cdot \delta_{ij}$ **Matrix notation:** $A^T \cdot B = C \cdot I$

Bi-orthogonal transforms are not necessarily orthogonal

Frequency / sequency of basis vectors: detailization levels

Discrete Cosine Transform (DCT)

(size 8, sorting by increasing frequency)



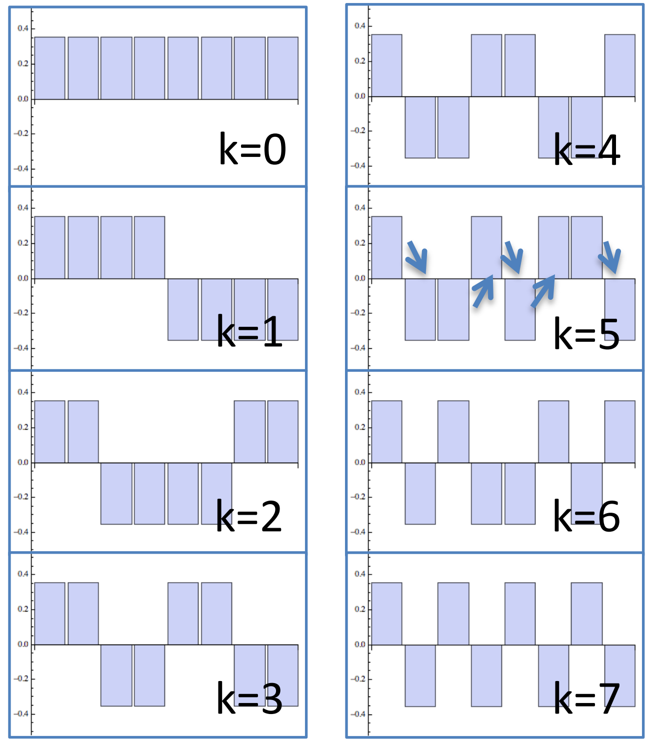
$$\vec{X} = A \cdot \vec{x},$$ Size: N

$$a[k,n] = \sqrt{\frac{2}{N}} C_k \cos\left\{(2n+1)\cdot\frac{\pi \cdot k}{2N}\right\},$$

$$b[n,k] = \sqrt{\frac{2}{N}} C_k \cos\left\{(2n+1)\cdot\frac{\pi \cdot k}{2N}\right\}$$

Walsh-Hadamard Transform (WHT):

(size 8, sorting by increasing sequency)



Sequency: analog of frequency for "binary" signals, # of "switches" between two states

$$\vec{X} = H \cdot \vec{x},$$ Size: N

$$H^{(2)} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$H^{(2M)} = \frac{1}{\sqrt{2}}\begin{pmatrix} H^{(M)} & H^{(M)} \\ H^{(M)} & -H^{(M)} \end{pmatrix}$$

- WHT per se is not extremely useful for natural images
- Used in video codecs (for hierarchical de-correlation)
- Observation: doubling of the basis vector size is very simple
- Allows lossless reconstruction

Location vs frequency: case of a graphical sound equalizer

Audio spectrum analyzer:
normal Fourier analysis is not always convenient!

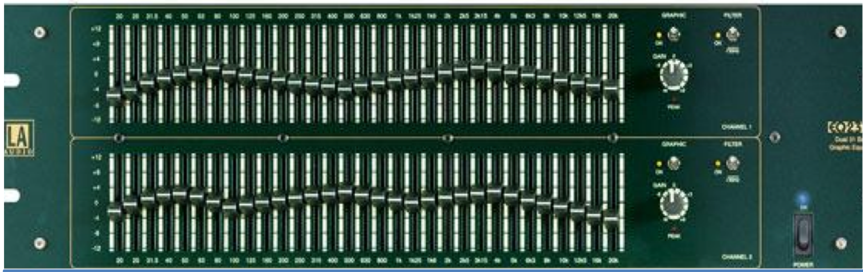
- Need the entire signal in advance to compute the coefficients of all frequencies
- The Fourier spectrum is static in time
- Typical signal elements (notes, words) last much longer than the typical signal oscillation

⇒ Need spectrum analysis in small signal chunks, i.e. in some "windows".

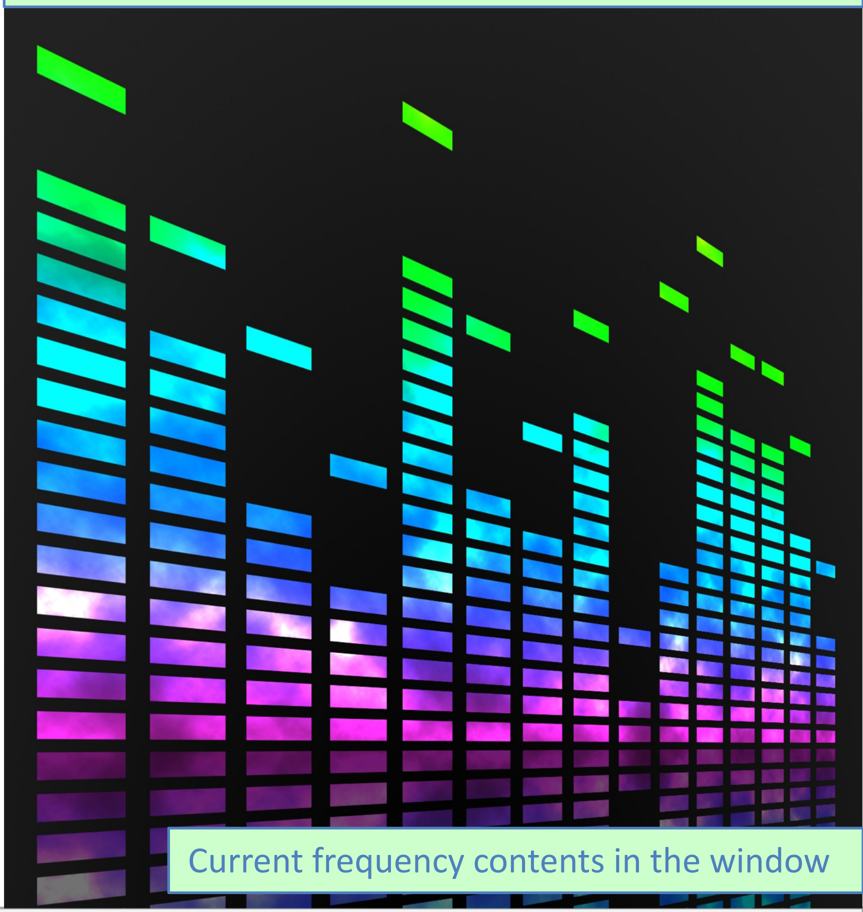
(*Have you ever seen a window specification for an equalizer?)

Sliding signal window

The result is time-dependent **AND** frequency-dependent. But: frequency and time position cannot be known exactly at the same time!



A trade-off between temporal and spectral accuracy

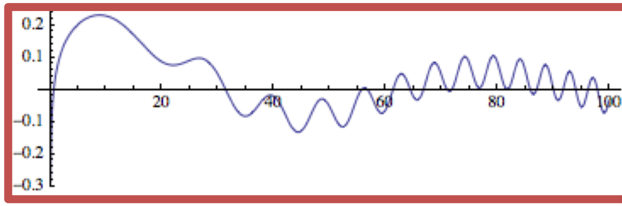


Current frequency contents in the window

Spatial vs frequency resolution

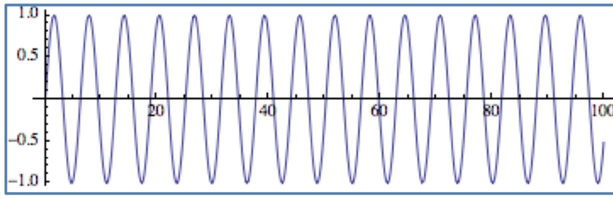
A complicated signal, contains multiple frequencies and localized features

Original signal:



Sample convolution values and their interpretation:

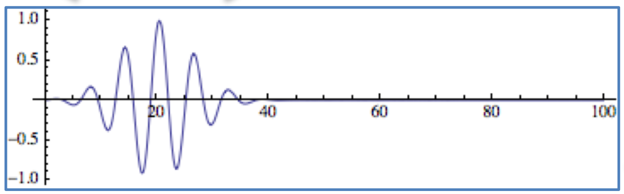
Standard FT kernel function:



42

- Content of frequency f_0 in the signal
- All spatial information is lost
- Need all signal values to compute
- Kernel defined by one parameter: f_0

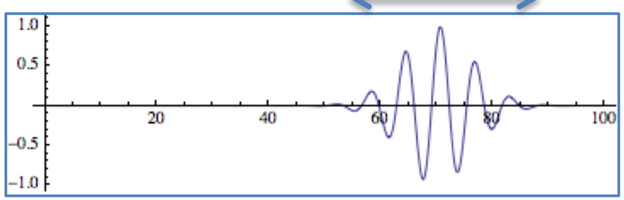
"Windowed Fourier", or "Short-Time Fourier" transform kernel at $x_0 = 20$:



42

- Content of freq.'s $f \approx f_0$ near $x \approx 20$
- Need only the signal values in window
- Kernel DFT is no longer a δ-function!

Another WFT kernel centered at $x_0 = 70$:

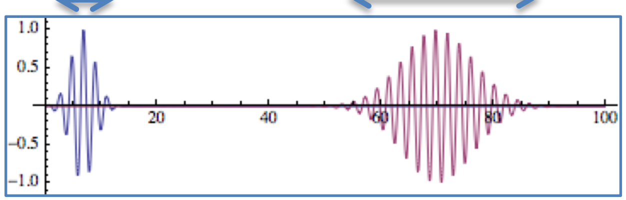


42

- Content of freq.'s $f \approx f_0$ near $x \approx 70$
- Analysis kernel shifted along the x-axis

What is better?

WFT at three times the frequency f_0:

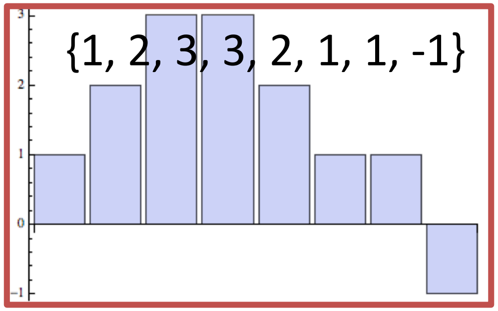


42

- Content of freq.'s $f \approx 3f_0$ for $x \approx 20$
- Choice: stretch the window or get more samples in the same window?
- Need at least parameters f_0 and x_0

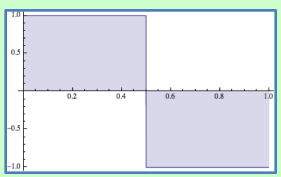
Discrete wavelet transform structure - example

Original signal x[n]:

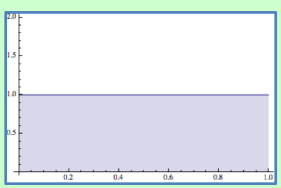


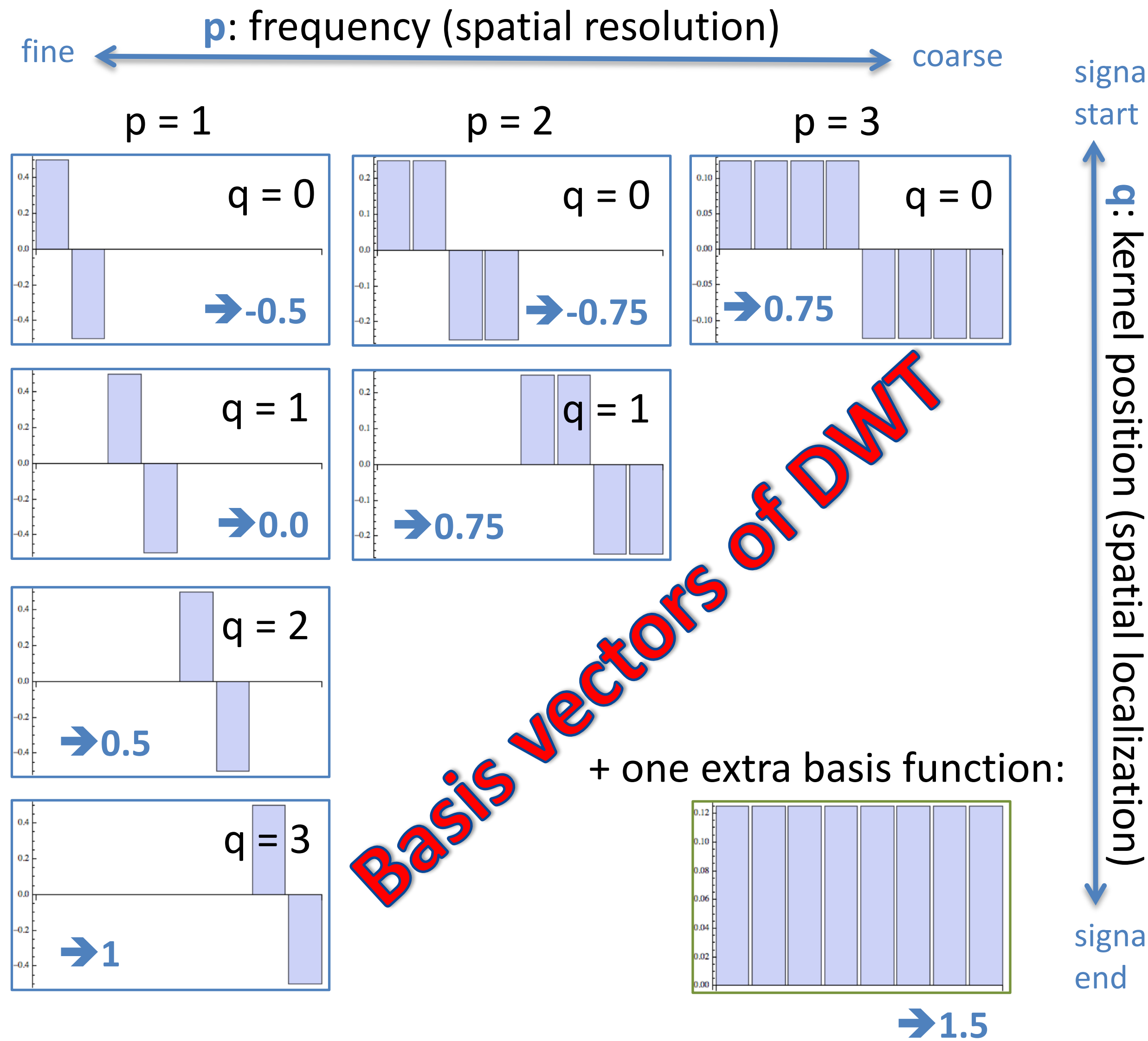
- Two transform indices: p and q
- Analysis: same basis vector, stretched according to p and shifted right by q signal widths
- Last step: compute mean value (i.e. use different kernel shape)

Wavelet function:



Scaling function:





Transformed signal X[k]: {-0.5, 0, 0.5, 1, -0.75, 0.75, 0.75, 1.5}

Discrete wavelet transform: notation

"Mother wavelet": $\psi(t)$

(scaling function may be computed from mother wavelet as the complement)

Transform kernel: $\psi_{p,q}(t) = \frac{1}{\sqrt{p}} \cdot \psi\left(\frac{t-q}{p}\right),$

Discrete scaling index p, shift q: $p = p_0^{-j}, q = k \cdot q_0 \cdot p_0^{-j}$

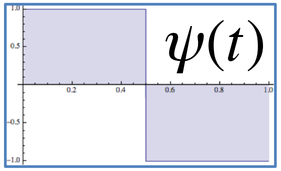
With $p_0 = 2, N = 8$, a DWT kernel is:

$$\psi_{p,q}[n] = \sqrt{2^j} \cdot \psi[n \cdot 2^j - k \cdot 8]$$

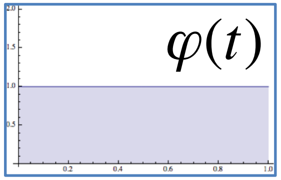
- May re-scale the direct and inverse transformation basis vectors to contain only integer-valued coefficients, and arrive at the result on the previous slide
- In principle, any compact-support "benign" function can be a wavelet (it can also be adapted to the signal!)
- Inverse transformation is computed as an inverse matrix,

$$W_B = W_A^{-1}$$

Wavelet function:



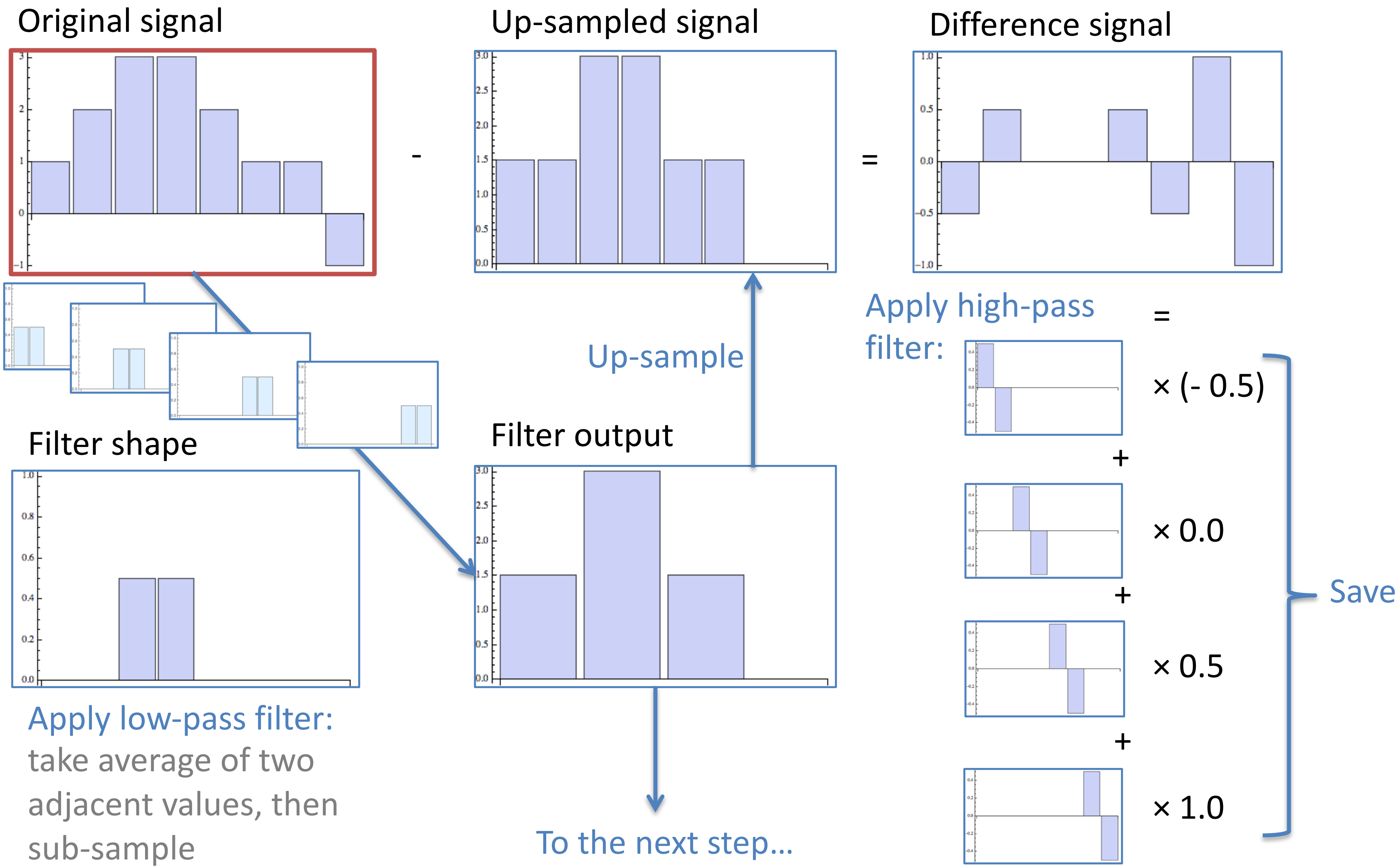
Scaling function:



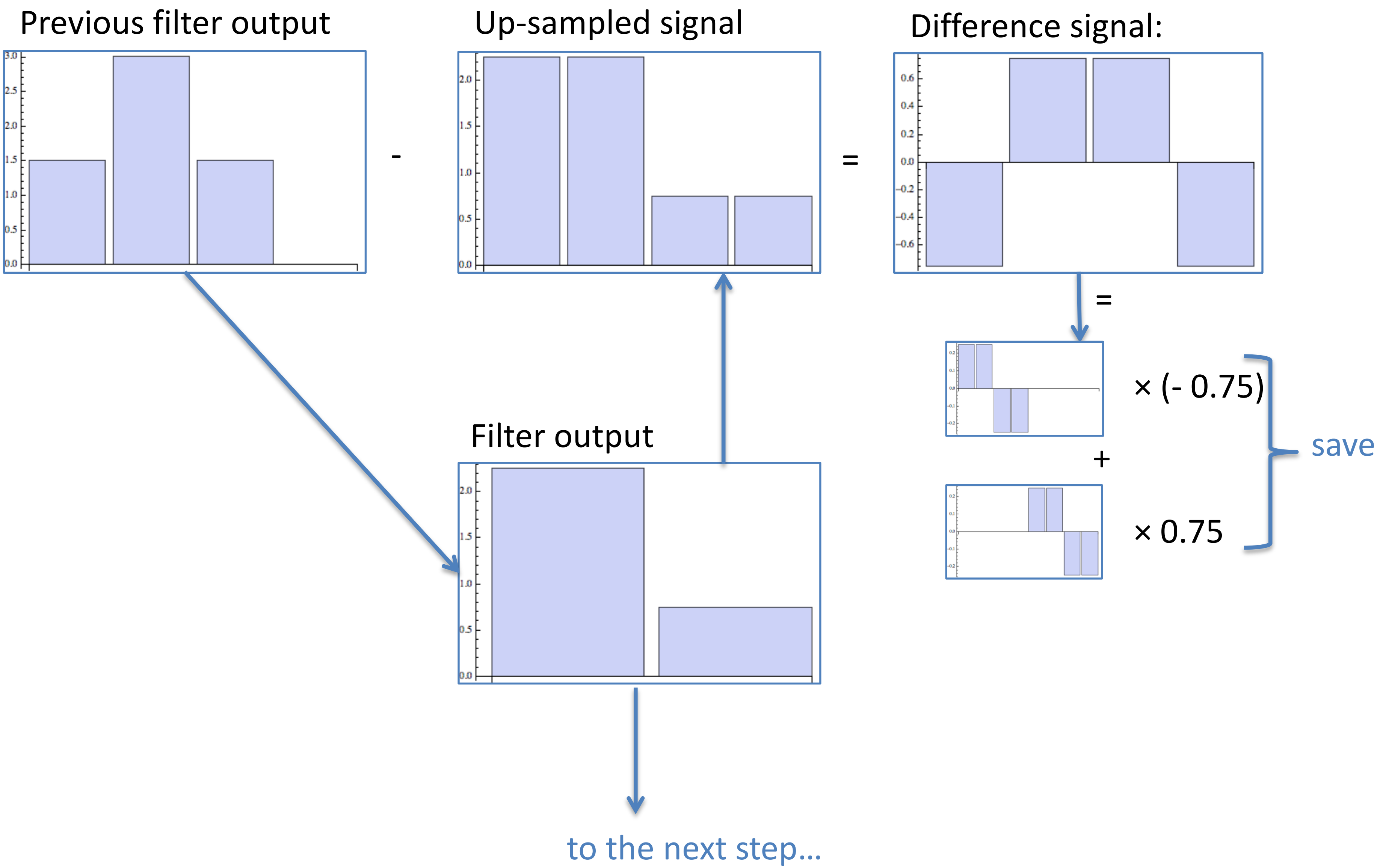
(Haar wavelet, [Haar 1910])

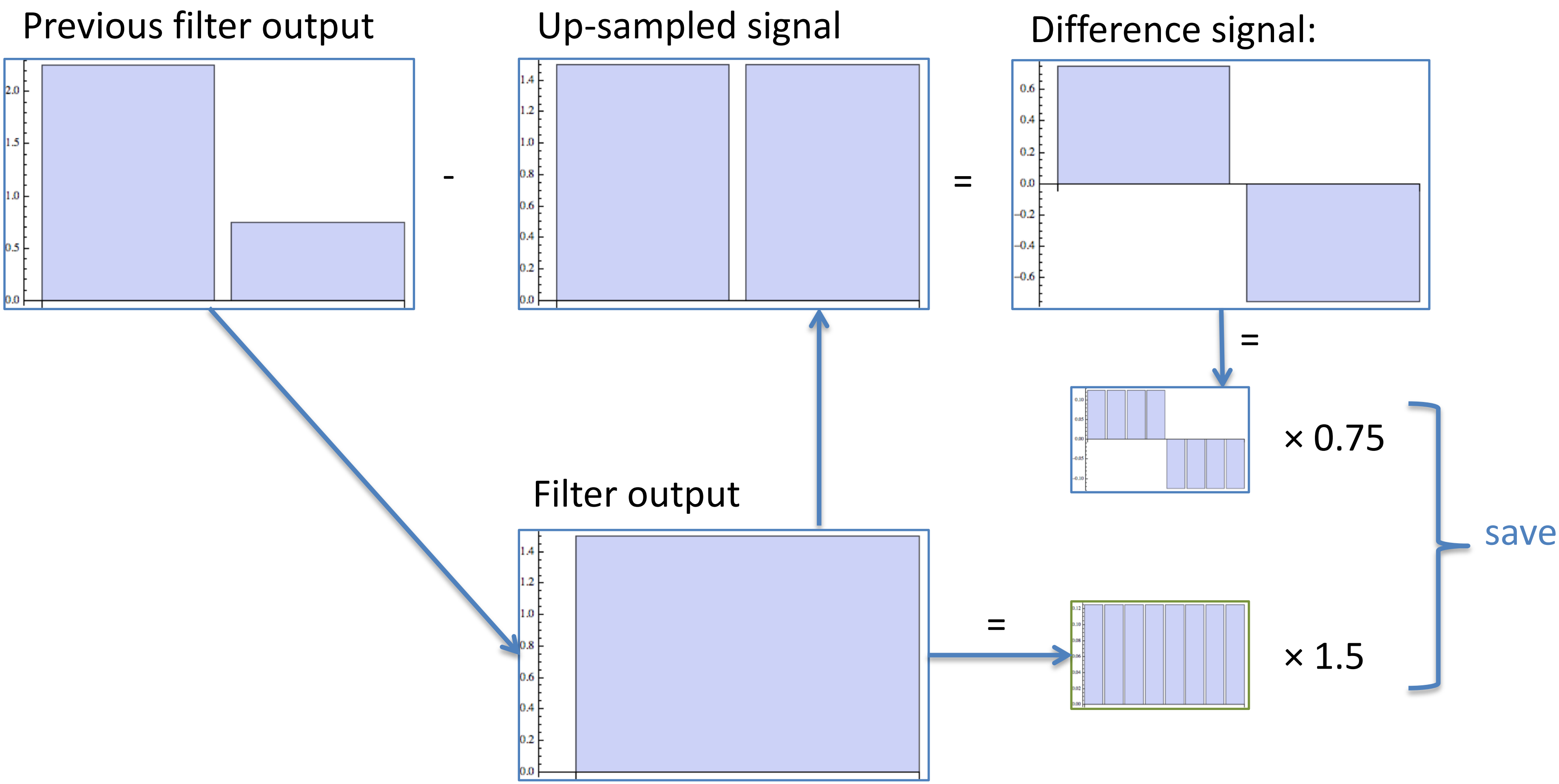
$$W_A^{(8)} = \begin{pmatrix} \varphi \\ \psi_{0,0} \\ \psi_{1,0} \\ \psi_{1,1} \\ \psi_{2,0} \\ \psi_{2,1} \\ \psi_{2,2} \\ \psi_{2,3} \end{pmatrix}$$

Wavelet transform: alternative derivation via filters



Wavelet transform: alternative derivation via filters





Transformed signal: {-0.5, 0, 0.5, 1, -0.75, 0.75, 0.75, 1.5}

Filter banks (FBs): more formal introduction

Common tool to study signals and filters

Original signal: $x[n], n \in Z$

Assume finite convergence radius, existence, etc.

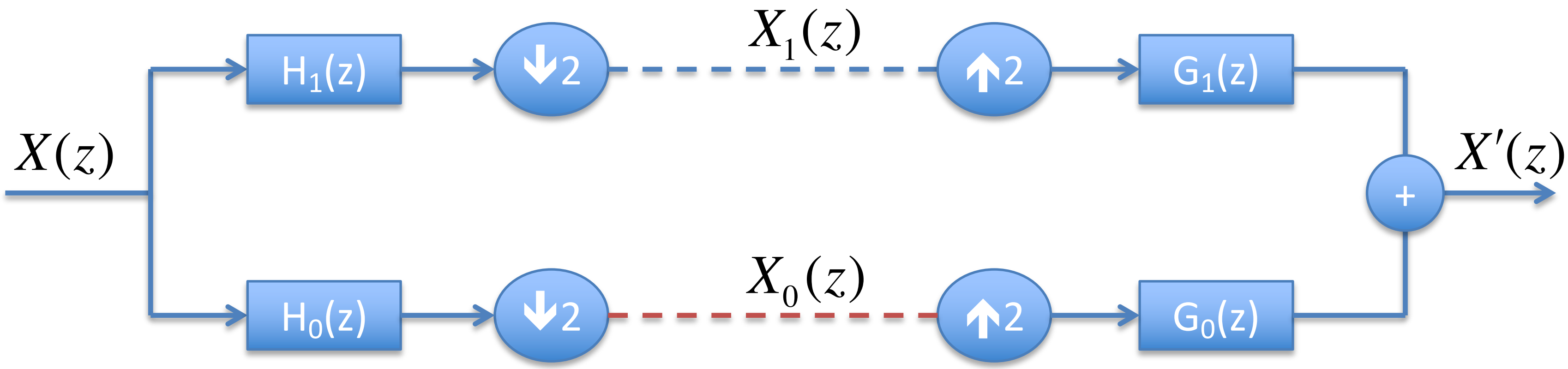
z-transform of a signal: $x[n] \to X(z), \quad X(z) = \sum_n x[n] \cdot z^{-n}$

Basic properties of the z-transform:

- Signal reflection: $y[n] = x[-n] \to X(z^{-1})$
- Alternating signs: $y[n] = (-1)^n x[n] \to X(-z)$
- Signal shift: $y[n] = x[n+l] \to X(z) \cdot z^l$
- Up-sampling: $y[n] = \begin{cases} x[n/L], n = m \cdot L, m \in Z \\ 0, otherwise \end{cases} \to X(z^L)$
- Sub-sampling: $y[n] = \frac{1}{M} \sum_{k=0}^{M-1} X(z^{1/M} \cdot e^{2\pi ik/M}) \to x[n \cdot M], M \in N$
- Convolution: $x[n] * y[n] = \sum_{k=-\infty}^{+\infty} x[k] \cdot y[n-k] \to X(z) \cdot Y(z)$

Filter applied to a signal!

2-channel analysis-synthesis filter bank



Filters: $h_0[n] \to H_0(z)$, etc.

Using properties of z-transform, we find:

$$X_1(z) = \frac{1}{2}\left[H_1(z^{1/2}) \cdot X(z^{1/2}) + H_1(-z^{1/2}) \cdot X(-z^{1/2})\right]$$

$$X_0(z) = \frac{1}{2}\left[H_0(z^{1/2}) \cdot X(z^{1/2}) + H_0(-z^{1/2}) \cdot X(-z^{1/2})\right]$$

$$X'(z) = G_1(z) \cdot X_1(z^2) + G_0(z) \cdot X_0(z^2)$$

Need **Perfect Reconstruction** (PR) of the input signal: allow at most a constant delay in the output,

$$X'(z) = X(z) \cdot z^{-l}$$

Contribution with the alternating signs contains completely foreign frequencies, must vanish!

After the substitution:

$$X'(z) = \overbrace{\frac{G_1(z)H_1(z) + G_0(z)H_0(z)}{2}}^{z^{-l}} \cdot X(z) + \overbrace{\frac{G_1(z)H_1(-z) + G_0(z)H_0(-z)}{2}}^{0} \cdot X(-z)$$

Orthogonal filter banks

Perfect Reconstruction conditions:

$$\frac{1}{2}\left(G_1(z)H_1(z)+G_0(z)H_0(z)\right)=z^{-l}$$

$$G_1(z)H_1(-z)+G_0(z)H_0(-z)=0$$

Second line is satisfied $\forall z$ iff:

$$\begin{cases} G_1(z)=P(z)H_0(-z) \\ H_1(-z)=-P(z)^{-1}G_0(z) \end{cases}$$

H_i, G_i non-recursive (FIR) filters,
➔P(z) may be at most a monomial:

$$P(z)=c\cdot z^{-m}, m\in Z$$

First PR condition:
(e.g., eliminate G_i)

$$H_0(-z)H_1(z)-H_0(z)H_1(-z)=\frac{2}{c}z^{m-l}$$

From the time-domain perspective: this is possible only for odd values of (m – l)

Conjugate Quadrature Filter bank (CQFB), [Smith, Barnell '84]:

Assume:

- Number N of impulse response coefficients in h_0 is even
- Let m – l = 1 – N, c = 2, m = 0, and also
- $H_1(z)=z^{1-N}H_0(-z^{-1})$

G_1: attenuated version of H_0 with alternating-sign values

Then:

$$H_0(-z)H_0(-z^{-1})+H_0(z)H_0(z^{-1})=1,$$

$$G_1(z)=2H_0(-z),$$

$$G_0(z)=2z^{1-N}H_0(z^{-1}).$$

Back to the time domain:

$$h_1[n]=(-1)^{N-1-n}h_0[N-1-n],$$

$$g_0[n]=2h_0[N-1-n],$$

$$g_1[n]=2(-1)^n h_0[n],$$

$$\sum_{k=-\infty}^{\infty}h_0[k]\cdot h_0[k-2n]=\begin{cases}1/2, n=0\\ 0, n\neq 0\end{cases}$$

Orthogonality:

- $h_0 \times h_1 = 0$,
- $g_0 \times g_1 = 0$.

N/2 conditions on N unknowns $h_0[n]$

Perfect Reconstruction conditions:

$$\frac{1}{2}\left(G_1(z)H_1(z)+G_0(z)H_0(z)\right)=z^{-l}$$

$$G_1(z)H_1(-z)+G_0(z)H_0(-z)=0$$

Second line satisfied $\forall z$ iff:

$$\begin{cases} G_1(z)=P(z)H_0(-z) \\ H_1(-z)=-P(z)^{-1}G_0(z) \end{cases}$$

H_i, G_i non-recursive (FIR) filters,
➔P(z) may be at most a monomial,

$$P(z)=c\cdot z^{-m}, m\in Z$$

First PR condition:
(this time, eliminate H_1)

$$H_0(z)G_0(z)-(-1)^m H_0(-z)G_0(-z)=2z^{-l}$$

Bi-orthogonal FBs: PR but not necessarily orthogonal

Assume l = 0 **, then** $H_0(z)G_0(z)+H_0(-z)G_0(-z)=2$ **, or** $\sum_{k=-\infty}^{\infty} h_0[k]g_0[2n-k]=\begin{cases}1, n=0\\0, n\neq 0\end{cases}$

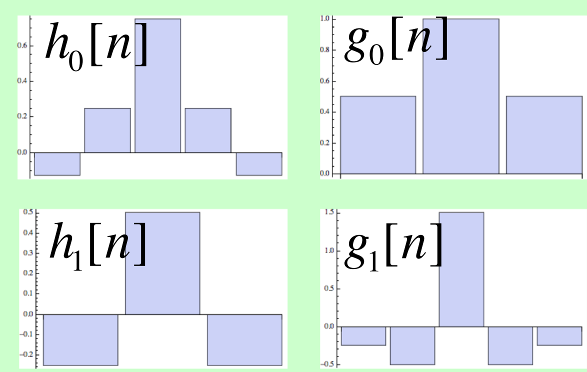
In addition, assume: $\sum_{k=-\infty}^{\infty} h_0[k]=1,\quad P(z)=2z^{-1}$

Cross-like relations between h and g

Then the filter impulse responses are:

$$\begin{cases} h_1[n]=-2^{-1}(-1)^n g_0[n+1], \\ g_1[n]=-2(-1)^n h_0[n-1]. \end{cases}$$

Example: 5/3 BO FB



More freedom allowed than with orthogonal filters!

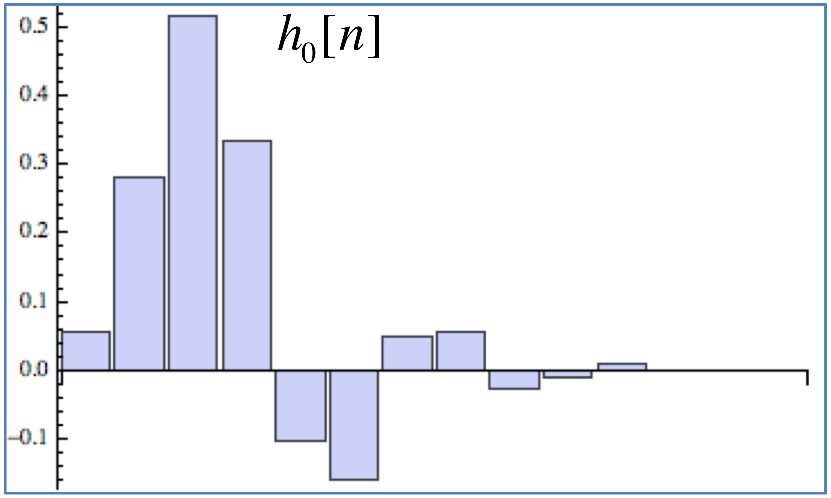
An orthogonal filter bank with impulse responses of length **N** has **N / 2** free parameters
One possible way to fix them:

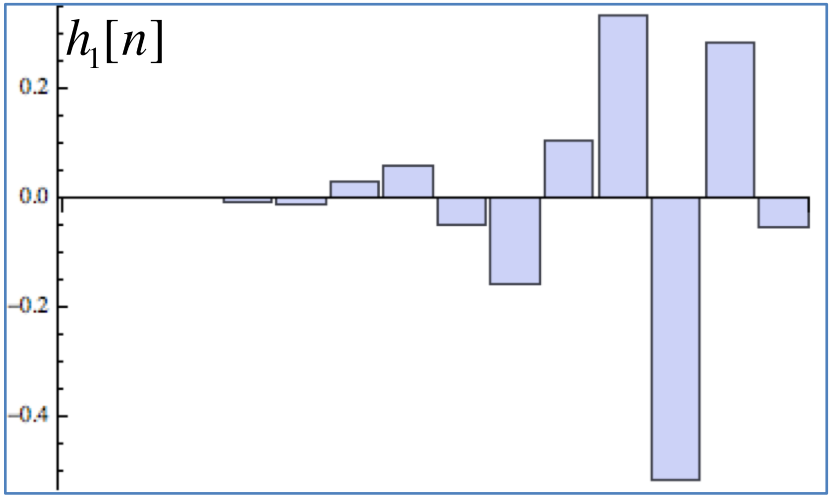
$$\sum_{n=-\infty}^{\infty} h_1[n] \cdot n^p = 0, p = 0, 1, \ldots, \frac{N}{2} - 1$$

- Resulting filters/wavelets: Daubechies family [Daubechies '88]
- With **N = 1**, equals Haar filter/wavelet

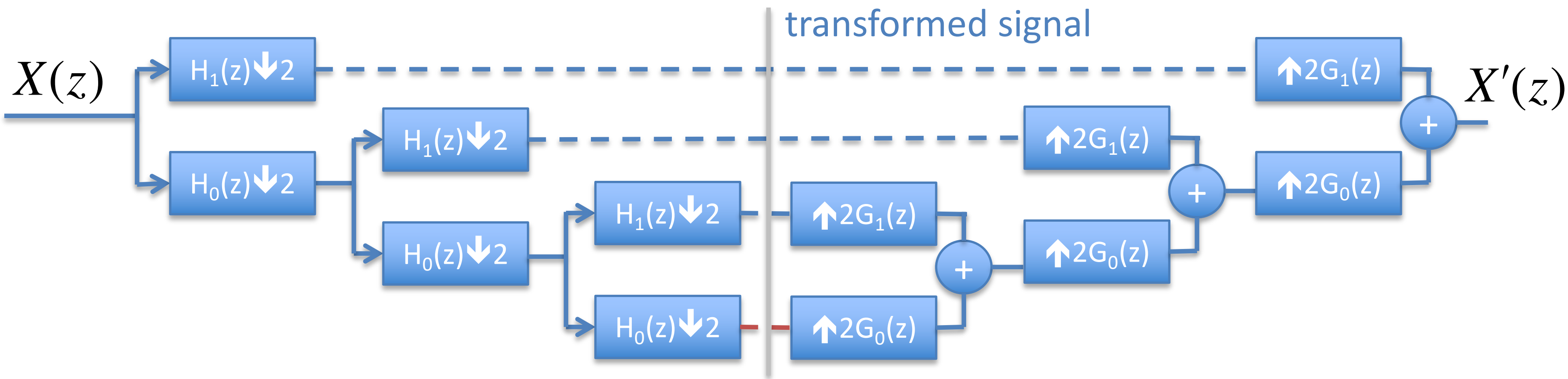
The more vanishing moments, the more "smoothing" action the high-pass filter has

Example: orthogonal Daubechies wavelet filter coefficients for **N / 2 = 7** (Daubechies7):





Cascading: octave-band filter banks (dyadic WT)



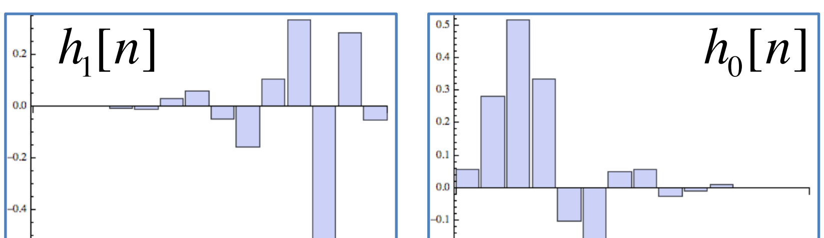
Extracting discrete wavelet functions:

- Set a single transformed coefficient to **1** and the remaining to **0**, measure the response
- The first coefficient corresponds to the smallest wavelet, equals the high-pass filter response
- The last coefficient corresponds to discrete scaling function

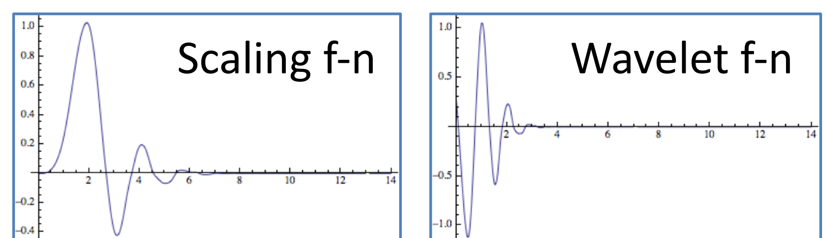
Bi-orthogonal wavelets: can switch analysis and synthesis filters, get dual wavelet basis

Continuous wavelet functions:

- Limit of the discrete wavelets after infinitely many cascading stages (i.e. the fixed point)



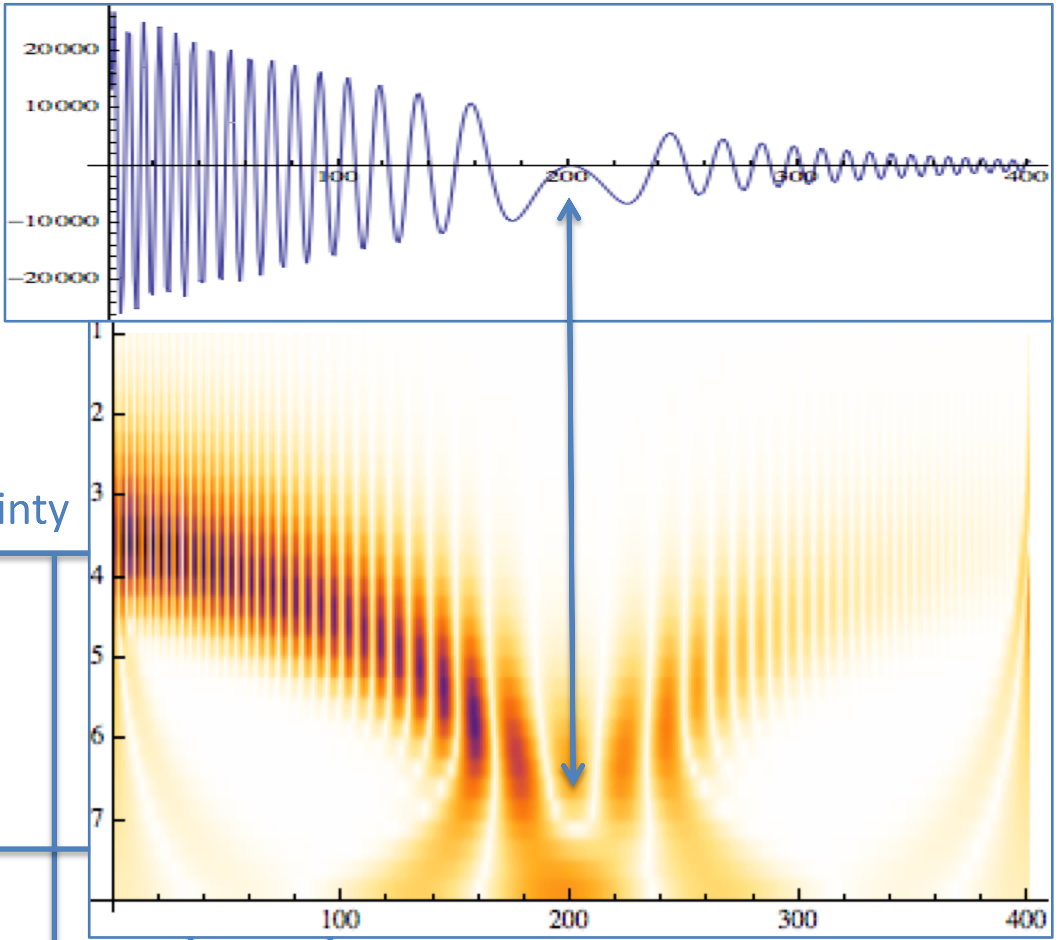
Daubechies7 analysis filters

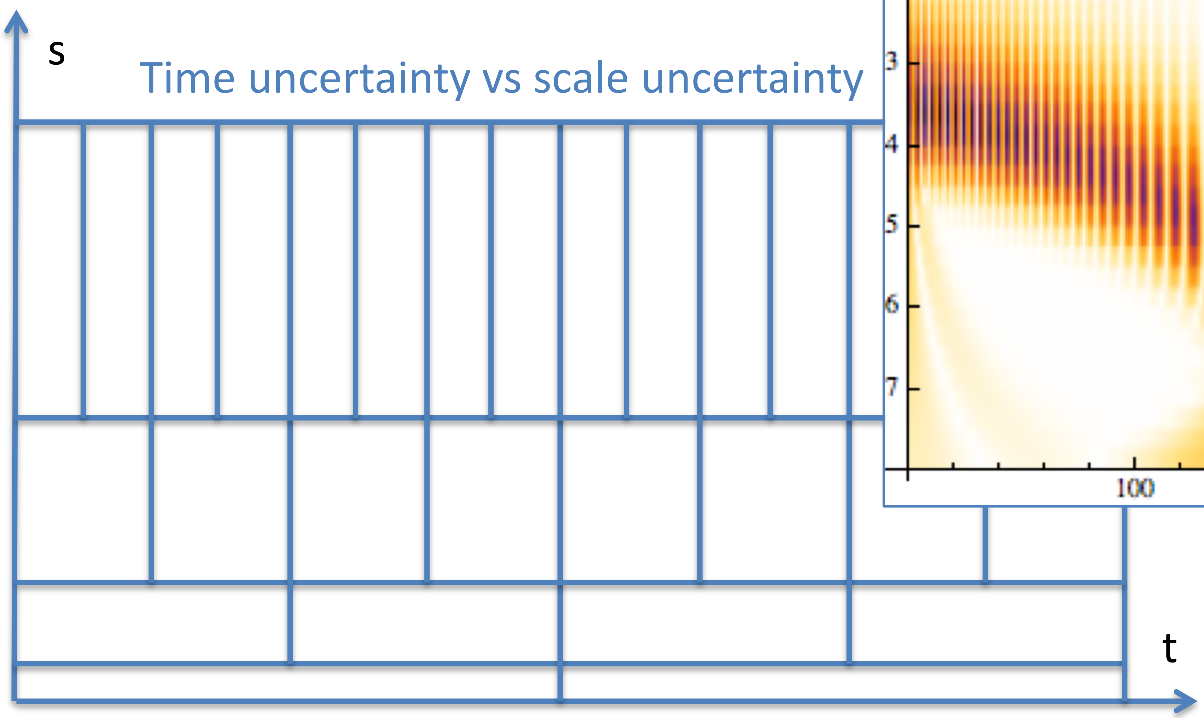


Daubechies7 wavelet

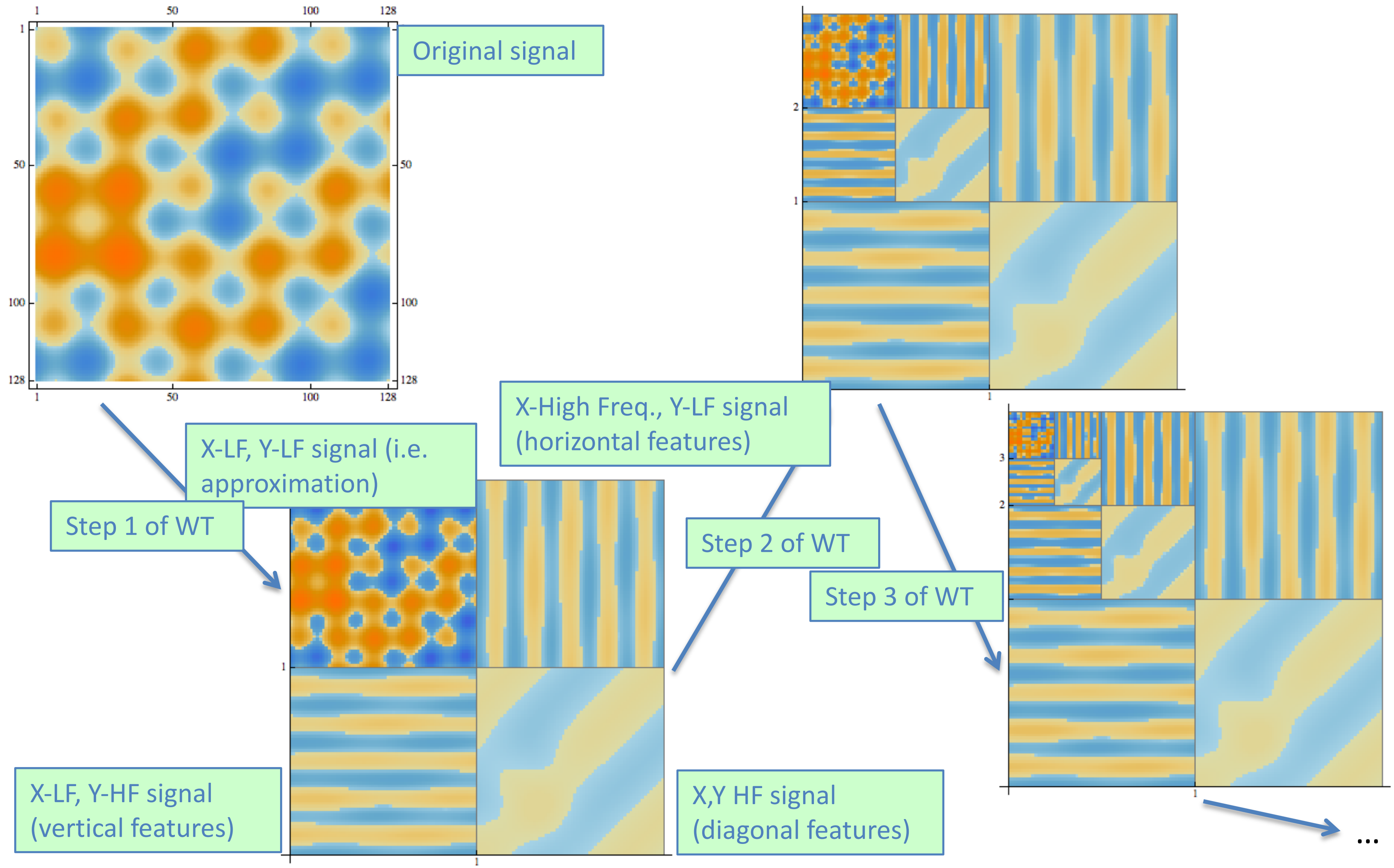
Time-scale plane representation (scalogram)

Example: signal scalogram (CWT, Haar wavelet)



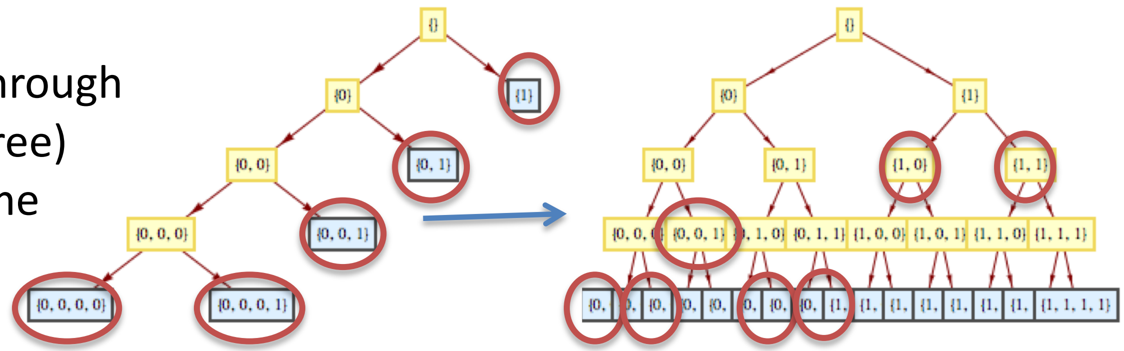


A separable transform: apply vertical and horizontal filter cascades



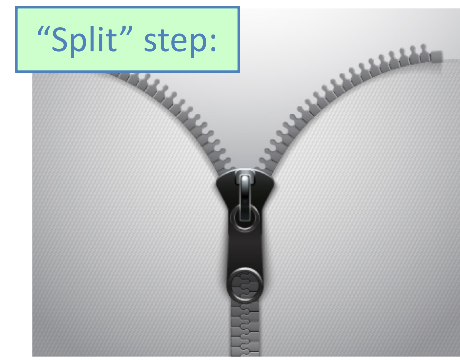
Wavelet packet transform:

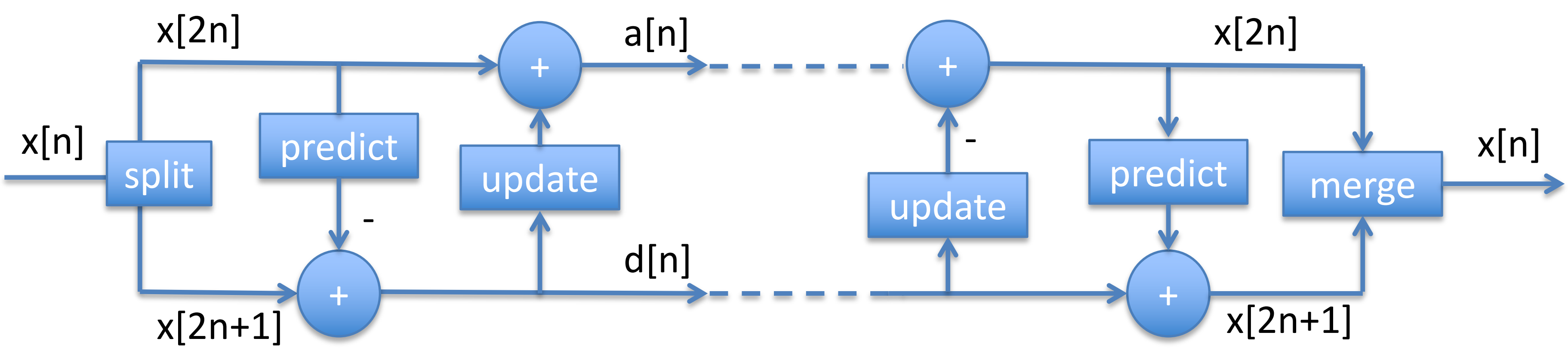
- Send the high-pass output also through the filter bank (i.e. build a filter tree)
- Select coefficients optimizing some metric (e.g. number of bits in the transformed representation)



Lifting schemes [Sweldens '95]:

- Model the filter action with a split-predict-update scheme
- Equivalent lifting schemes exist for any PR filter bank, at least 2 times faster (reduce redundant work done in FB)
- Completely understood for CDF wavelets
- Fast integer-only lifting schemes allow lossless DWT!

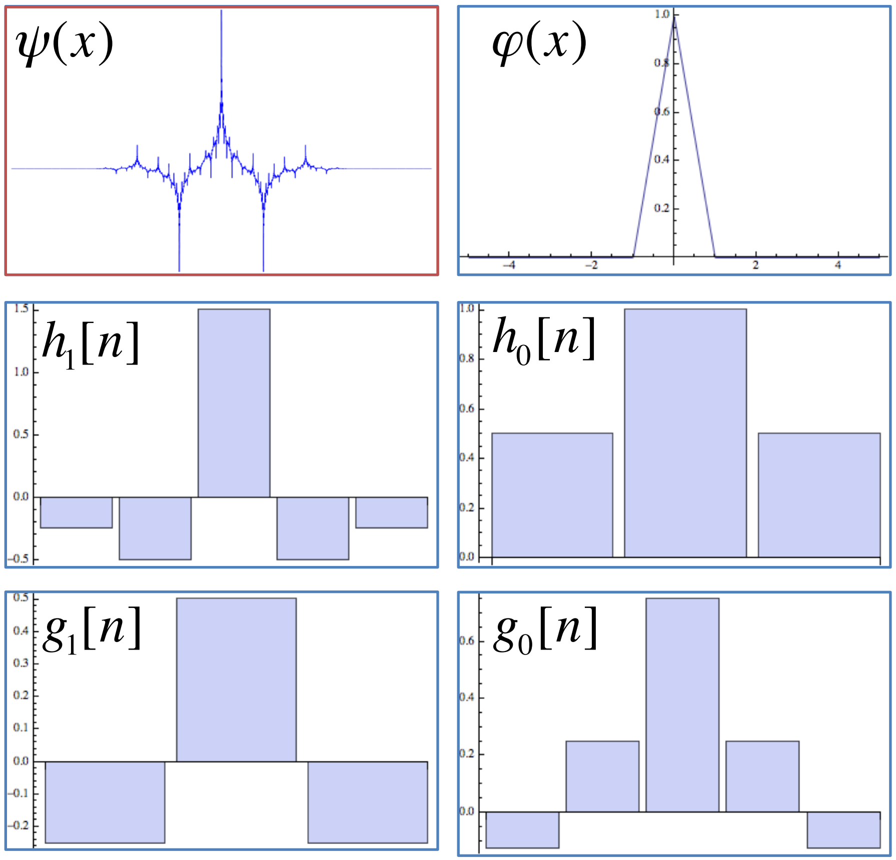


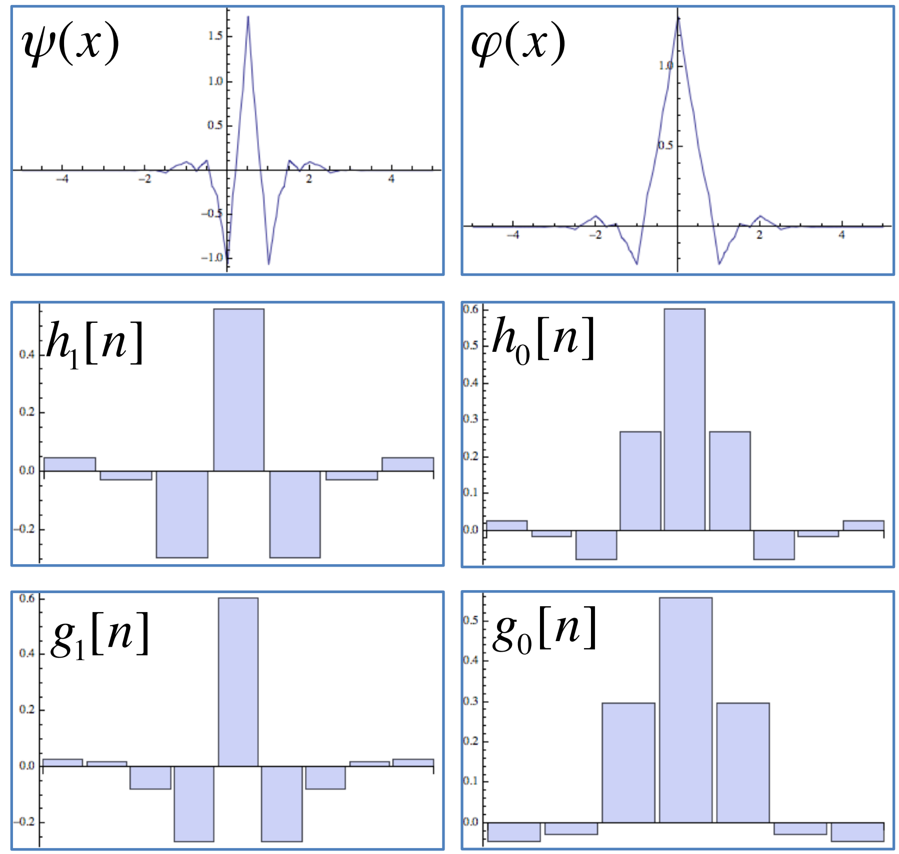


Practical use of DWT – JPEG2000 standard

Lossless compression:
- Bi-orthogonal Cohen-Daubechies-Feauveau (CDF) 5/3 (aka LeGall) wavelet
- Integer (simple rational) filter coefficients
- Integer lifting-based computation

Lossy compression:
- CDF 9/7 wavelet (BO, symmetric)
- DWT followed by non-linear quantization
- Resistant to noise in DWT coefficients
- 4 and 4 vanishing moments (maximum)
- Small support (few non-zero filter values)





Summary – discrete wavelet transform

- Discrete wavelet transform (DWT) is a discrete linear transform of discrete signals. (☺)
- Wavelets (basis vectors) may be selected arbitrarily acc. to some rules: DWT is not unique.
- DWT coefficients usually describe time- and frequency-localized features.
- Direct computation of DWT coefficients is not very efficient. A better method: use filter banks / cascades / trees. Efficient, applicable to continuous signals!
- Filters used in FBs must be perfectly reconstructing for the transform to be reversible.
- There are multiple methods to generate/choose suitable PR filters. Some better-known recipes: conjugate quadrature filters, bi-orthogonal filters. Filters with vanishing moments (e.g. Daubechies family) "smooth" the signal.
- Given a set of PR filters, respective wavelet functions are difficult to recover (even plot).
- Best practical implementations rely on lifting schemes (cleverly avoid extra work).
- Some FBs implementing DWT allow lossless reconstruction by using integer-valued filters.